

Multi-Agent Distributed Optimization via Inexact Consensus ADMM

Tsung-Hui Chang^{*}, Mingyi Hong[†] and Xiangfeng Wang[‡]

Abstract

Multi-agent distributed consensus optimization problems arise in many signal processing applications. Recently, the alternating direction method of multipliers (ADMM) has been used for solving this family of problems. ADMM based distributed optimization method is shown to have faster convergence rate compared with classic methods based on consensus subgradient, but can be computationally expensive, especially for problems with complicated structures or large dimensions. In this paper, we propose low-complexity algorithms that can reduce the overall computational cost of consensus ADMM by an order of magnitude for certain large-scale problems. Central to the proposed algorithms is the use of an inexact step for each ADMM update, which enables the agents to perform cheap computation at each iteration. Our convergence analyses show that the proposed methods can converge well under mild conditions. Numerical results show that the proposed algorithms offer considerably lower computational complexity at the expense of extra communication overhead, demonstrating potential for certain big data scenarios where communication between agents can be implemented cheaply.

Keywords— Distributed optimization, ADMM, Consensus

EDICS: OPT-DOPT, MLR-DIST, NET-DISP, SPC-APPL.

The work of Tsung-Hui Chang is supported by National Science Council, Taiwan (R.O.C.), under Grant NSC 102-2221-E-011-005-MY3. The work of Xiangfeng Wang is supported by Doctoral Fund of the Ministry of Education, China, No.20110091110004. Part of this work was submitted to IEEE ICASSP 2014.

^{*}Tsung-Hui Chang is the corresponding author. Address: Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan, (R.O.C.). E-mail: tsunghui.chang@ieee.org.

[†]Mingyi Hong is with Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, E-mail: mhong@umn.edu

[‡]Xiangfeng Wang is with Department of Mathematics, Nanjing University, Nanjing 210093, P. R. China, E-mail: wangxf@smail.nju.edu.cn

I. INTRODUCTION

We consider a network with multiple agents, for example a sensor network with distributed sensor nodes, a data cloud network with distributed database servers, a communication network with distributed base stations (mobile users) or even a computer system with distributed microprocessors. We assume that the network consists of N agents who collaborate with each other to accomplish certain tasks. For example, distributed database servers may cooperate for data mining or for parameter learning in order to fully exploit the data collected from individual servers [1]. Another example arises from big data applications [2], where a computation task may be executed by collaborative distributed microprocessors with individual memories and storage spaces [3], [4]. Many of the distributed optimization tasks, such as those described above, can be cast as a generic optimization problem of the following form

$$(P1) \quad \min_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^N \phi_i(\mathbf{x}) \quad (1)$$

where $\mathbf{x} \in \mathbb{R}^K$ is the decision variable, $\mathcal{X} \subseteq \mathbb{R}^K$ is the feasible set of \mathbf{x} , and $\phi_i : \mathbb{R}^K \rightarrow \mathbb{R}$ is the cost function associated with agent i . Here the function ϕ_i is composed of a smooth component f_i and a non-smooth component g_i , i.e.,

$$\phi_i(\mathbf{x}) = f_i(\mathbf{A}_i \mathbf{x}) + g_i(\mathbf{x}),$$

where $\mathbf{A}_i \in \mathbb{R}^{M \times K}$ is some data matrix not necessarily of full rank. Such model is common in practice: the smooth component usually represents the cost function to be minimized, while the non-smooth component is often used for regularization purposes [5].

In the setting of distributed optimization, it is commonly assumed that each agent i only has knowledge about the local information f_i , g_i and \mathbf{A}_i . The challenge is to obtain, for each agent in the system, the optimal \mathbf{x} of (P1) using only local information and messages exchanged with neighbors [6]–[9].

Problem (P1) is closely related to the following problem

$$(P2) \quad \min_{\mathbf{x}_1 \in \mathcal{X}_1, \dots, \mathbf{x}_N \in \mathcal{X}_N} \sum_{i=1}^N \phi_i(\mathbf{x}_i) \quad \text{s.t.} \quad \sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i = \mathbf{q}, \quad (2)$$

where $\mathbf{E}_i \in \mathbb{R}^{M \times K}$, $\mathbf{q} \in \mathbb{R}^M$ and $\mathcal{X}_i \subseteq \mathbb{R}^K$. Unlike (P1), in (P2), each agent i owns a local control variable \mathbf{x}_i , and these variables are coupled together through the linear constraint. Examples of (P2) include the basis pursuit (BP) problem [10], [11], the network flow control problem [12] and interference management problem in communication networks [13]. To relate (P2) with (P1), let $\mathbf{y} \in \mathbb{R}^M$ be the Lagrange dual variable associated with the linear constraint $\sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i = \mathbf{q}$. The Lagrange dual problem of (P2) can be written as

$$\min_{\mathbf{y} \in \mathbb{R}^M} \sum_{i=1}^N \left(\varphi_i(\mathbf{y}) + \frac{1}{N} \mathbf{y}^T \mathbf{q} \right) \quad (3)$$

where

$$\varphi_i(\mathbf{y}) = \max_{\mathbf{x}_i \in \mathcal{X}_i} -\phi_i(\mathbf{x}_i) - \mathbf{y}^T \mathbf{E}_i \mathbf{x}_i, \quad i = 1, \dots, N. \quad (4)$$

Problem (3) thus has the same form as (P1). Given the optimal \mathbf{y} of (3) and assuming that (P2) has a zero duality gap [14], each agent i can obtain the associated optimal variable \mathbf{x}_i by solving (4). Therefore, a distributed optimization method that can solve (P1) may also be used for (P2) through solving (3).

There is an extensive literature on distributed consensus optimization methods, such as the consensus subgradient methods; see [6], [7] and the recent developments in [8], [9], [15], [16]. The consensus subgradient methods are appealing owing to their simplicity and the ability to handle a wide range of problems. However, the convergence of the consensus subgradient methods are usually slow.

Recently, the alternating direction method of multipliers (ADMM) [17] has become popular for solving problems with forms of (P1) and (P2) in a distributed fashion. In [13], distributed transmission designs for multi-cellular wireless communications were developed based on ADMM. In [18], several ADMM based distributed optimization algorithms were developed for solving the sparse LASSO problem [19]. In [11], using a different consensus formulation from [18] and assuming the availability of a certain coloring scheme for the graph, ADMM is applied to solving the BP problem [10] for both row partitioned and column partitioned data models [15]. In [20], the methodologies proposed in [11] are extended to handling a more general class of problems with forms of (P1) and (P2). The fast practical performance of ADMM is corroborated by its nice theoretical property. In particular, ADMM was found to converge linearly for a large class of problems [21], [22], meaning a certain optimality measure can decrease by a constant fraction in each iteration of the algorithm. In [23], such fast convergence rate has also been built for the distributed method in [18].

It is important to note that existing ADMM based algorithms can be readily used to solve problems (P1) and (P2). For example, by applying the consensus formulation proposed in [18] and ADMM to (P1), a fully parallelized distributed optimization algorithm can be obtained (where the agents update their variables in a fully parallel manner), which we refer to as the consensus ADMM (C-ADMM). To solve (P2), the same consensus formulation and ADMM can be used on its Lagrange dual (3), which leads to a distributed algorithm different from that in [11], referred to as the dual consensus ADMM (DC-ADMM). The main drawback of these algorithms lies in the fact that each agent needs to repeatedly

solve certain subproblems to *global optimality*. This can be computationally demanding, especially when the cost functions f_i 's have complicated structures or when the problem size is large [2]. If a low-accuracy suboptimal solution is used for these subproblems instead, the convergence is no longer guaranteed.

The main objective of this paper is to study algorithms that can significantly reduce the computational burden for the agents. In particular, we propose two algorithms, named the inexact consensus ADMM (IC-ADMM) and the inexact dual consensus ADMM (IDC-ADMM), both of which allow the agents to perform a single proximal gradient (PG) step [24] at each iteration. The benefit of the proposed approach lies in the fact that the PG step is usually simple, especially when g_i 's are structured sparse promoting functions [5], [24]. Notably, the cheap iterations of the proposed algorithms is made possible by *inexactly* solving the subproblems arising in C-ADMM and DC-ADMM, in a way that is not known in the ADMM or consensus literature. For example, in IC-ADMM, the proposed method approximates the smooth functions f_i 's in C-ADMM, which is very different from the known inexact ADMM methods [25], [26], which only approximate the quadratic penalty (thus does not always result in cheap PG steps). We summarize our main contributions below.

- For (P1), we propose an IC-ADMM method for reducing the computational complexity of C-ADMM. Conditions for global convergence of IC-ADMM are analyzed. Moreover, we show that IC-ADMM converges linearly, under similar conditions as in [23].
- For (P2), we propose a DC-ADMM method which, unlike the methods in [11], [20], can globally converge without any bipartite network or strongly convex ϕ_i 's.
- We further propose an IDC-ADMM method for reducing the computational burden of DC-ADMM. Conditions for global (linear) convergence are presented.

Numerical examples for solving distributed sparse logistic regression problems [27] will show that the proposed IC-ADMM and IDC-ADMM methods converge much faster than the consensus subgradient method [6]. Further, compared with the original C-ADMM and DC-ADMM, the proposed method can reduce the overall computational cost by an order of magnitude, despite using larger numbers of iterations.

The paper is organized as follows. Section II presents the applications and assumptions. The C-ADMM and IC-ADMM are presented in Section III; while DC-ADMM and IDC-ADMM are presented in Section IV. Numerical results are given in Section V and conclusions are drawn in Section VI.

II. APPLICATIONS AND NETWORK MODEL

A. Application to Data Regression

As discussed in Section I, (P1) and (P2) arise in many problems in sensor networks, data networks and machine learning tasks. Here let us focus on the classical regression problems. We consider a general formulation that incorporates the LASSO [18] and logistic regression (LR) [27] as special instances. Let $\mathbf{A} \in \mathbb{R}^{\bar{M} \times \bar{K}}$ denote a regression data matrix. For a row partitioned data (RPD) model [11, Fig. 1], [15], the distributed regression problem is given by

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{i=1}^N \Psi_i(\mathbf{x}; \mathbf{A}_i, \mathbf{b}_i), \quad (5)$$

where $\Psi_i(\mathbf{x}; \mathbf{A}_i, \mathbf{b}_i)$ is the cost function defined on the local regression data $\mathbf{A}_i \in \mathbb{R}^{(\bar{M}/N) \times \bar{K}}$ (which is the i th-row-block submatrix of \mathbf{A} , if a uniform partition is assumed) and a local response signal $\mathbf{b}_i \in \mathbb{R}^{\bar{M}}$. For example, the LASSO problem has $\Psi_i(\mathbf{x}; \mathbf{A}_i, \mathbf{b}_i) = \|\mathbf{b}_i - \mathbf{A}_i \mathbf{x}\|_2^2 + g_i(\mathbf{x})$. Similarly, for the LR problem, one has

$$\Psi_i(\mathbf{x}; \mathbf{A}_i, \mathbf{b}_i) = \sum_{m=1}^{\bar{M}/N} \log(1 + \exp(-b_{im} \mathbf{a}_{im}^T \mathbf{x})) + g_i(\mathbf{x}), \quad (6)$$

where $\mathbf{A}_i = [\mathbf{a}_{i1}, \dots, \mathbf{a}_{i(\bar{M}/N)}]^T$ contains \bar{M}/N training data vectors and $b_{im} \in \{\pm 1\}$ are binary labels for the training data. It is clear that (5) has the same form as (P1).

For the column partitioned data (CPD) model [11, Fig. 1], [15], the distributed regression problem is formulated as

$$\min_{\mathbf{x}_1 \in \mathcal{X}_1, \dots, \mathbf{x}_i \in \mathcal{X}_i} \sum_{i=1}^N \Psi_i(\mathbf{x}_i; \mathbf{E}_i, \mathbf{b}), \quad (7)$$

where the response signal \mathbf{b} is known to all agents while each agent i has a local regression variable $\mathbf{x}_i \in \mathbb{R}^{\bar{K}/N}$ and local regression data matrix $\mathbf{E}_i = [\mathbf{e}_{i1}, \dots, \mathbf{e}_{i\bar{M}}]^T \in \mathbb{R}^{\bar{M} \times (\bar{K}/N)}$ (which is the i th-column-block submatrix of \mathbf{A}). For example, the LR problem has $\Psi_i(\mathbf{x}_i; \mathbf{E}_i, \mathbf{b}) = \sum_{m=1}^{\bar{M}} \log(1 + \exp(-b_m \sum_{i=1}^N \mathbf{e}_{im}^T \mathbf{x}_i)) + g_i(\mathbf{x}_i)$. By introducing a slack variable $\mathbf{z} \triangleq \sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i$, the CPD LR problem can be reformulated as

$$\begin{aligned} \min_{\substack{\mathbf{x}_1 \in \mathcal{X}_1, \dots, \mathbf{x}_N \in \mathcal{X}_N, \\ \mathbf{z} \in \mathbb{R}^{\bar{M}}}} & \sum_{m=1}^{\bar{M}} \log(1 + \exp(-b_m z_m)) + \sum_{i=1}^N g_i(\mathbf{x}_i) \\ \text{s.t.} & \sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i - \mathbf{z} = \mathbf{0}, \end{aligned} \quad (8)$$

which is an instance of (P2). In Section V, we will primarily test our algorithms on the RPD and CPD regression problems.

B. Network Model and Assumptions

Let a graph \mathcal{G} denote a multi-agent network, which contains a node set $V = \{1, \dots, N\}$ and an edge set \mathcal{E} . An edge $(i, j) \in \mathcal{E}$ if and only if agent i and agent j can communicate with each other (i.e., neighbors). The edge set \mathcal{E} defines an adjacency matrix $\mathbf{W} \in \{0, 1\}^{N \times N}$, where $[\mathbf{W}]_{i,j} = 1$ if $(i, j) \in \mathcal{E}$ and $[\mathbf{W}]_{i,j} = 0$ otherwise. In addition, one can define an index subset $\mathcal{N}_i = \{j \in V \mid (i, j) \in \mathcal{E}\}$ for the neighbors of each agent i , and a degree matrix $\mathbf{D} = \text{diag}\{|\mathcal{N}_1|, \dots, |\mathcal{N}_N|\}$ (a diagonal matrix).

We make the following assumptions on \mathcal{G} and problems (P1) and (P2).

Assumption 1 *The network graph \mathcal{G} is connected.*

Assumption 1 implies that any two agents in the network can always influence each other in the long run. We also have the following assumptions on problems (P1) and (P2).

Assumption 2 (a) (P1) is a convex problem, i.e., ϕ_i 's are proper closed convex functions in \mathbf{x} and \mathcal{X} is a closed convex set. Moreover, strong duality holds for (P1).

(b) Problem (P2) is a convex problem, i.e., each ϕ_i is a proper closed convex in \mathbf{x}_i and \mathcal{X}_i is a closed convex set, for all $i \in V$. Moreover, strong duality holds for (P2).

Assumption 3 For all $i \in V$, the smooth function f_i is strongly convex, i.e., there exists some $\sigma_{f,i}^2 > 0$ such that

$$(\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x}))^T (\mathbf{y} - \mathbf{x}) \geq \sigma_{f,i}^2 \|\mathbf{y} - \mathbf{x}\|_2^2 \quad \forall \mathbf{y}, \mathbf{x} \in \mathbb{R}^M.$$

Moreover, f_i has Lipschitz continuous gradients, i.e., there exists some $L_{f,i} > 0$ such that

$$\|\nabla f_i(\mathbf{y}) - \nabla f_i(\mathbf{x})\|_2 \leq L_{f,i} \|\mathbf{y} - \mathbf{x}\|_2 \quad \forall \mathbf{y}, \mathbf{x} \in \mathbb{R}^M. \quad (9)$$

Note that, even under Assumption 3, $\phi_i(\mathbf{x}) = f_i(\mathbf{A}_i \mathbf{x}) + g_i(\mathbf{x})$ is not necessarily strongly convex in \mathbf{x} since the matrix \mathbf{A}_i can be fat and rank deficient. Both the LASSO problem [18] and the LR function in (6) satisfy Assumption 3¹.

III. DISTRIBUTED CONSENSUS ADMM

In Section III-A, we briefly review the original C-ADMM [18] for solving (P1). In Section III-B, we propose an computationally efficient inexact C-ADMM method.

¹The logistic regression function in (6) is strongly convex given that \mathbf{x} lies in a compact set.

A. Review of C-ADMM

Under Assumption 1, (P1) can be equivalently written as

$$\min_{\mathbf{x}_1 \in \mathcal{X}, \dots, \mathbf{x}_N \in \mathcal{X}, \{\mathbf{t}_{ij}\}} \sum_{i=1}^N \phi_i(\mathbf{x}_i) \quad (10a)$$

$$\text{s.t. } \mathbf{x}_i = \mathbf{t}_{ij} \quad \forall j \in \mathcal{N}_i, i \in V, \quad (10b)$$

$$\mathbf{x}_j = \mathbf{t}_{ij} \quad \forall j \in \mathcal{N}_i, i \in V, \quad (10c)$$

where $\{\mathbf{t}_{ij}\}$ are slack variables. According to (10), each agent i can optimize its local function $f_i(\mathbf{A}_i \mathbf{x}_i) + g_i(\mathbf{x}_i)$ with respect to a local copy of \mathbf{x} , i.e., \mathbf{x}_i , under the consensus constraints in (10b) and (10c). In [18], ADMM is employed to solve (10) in a distributed manner. Let $\{\mathbf{u}_{ij}\}$ and $\{\mathbf{v}_{ij}\}$ denote the dual variables associated with constraints (10b) and (10c), respectively. According to [18], ADMM leads to the following iterative updates at each iteration k :

$$\mathbf{u}_{ij}^{(k)} = \mathbf{u}_{ij}^{(k-1)} + \frac{c}{2}(\mathbf{x}_i^{(k-1)} - \mathbf{x}_j^{(k-1)}) \quad \forall j \in \mathcal{N}_i, i \in V, \quad (11a)$$

$$\mathbf{v}_{ij}^{(k)} = \mathbf{v}_{ij}^{(k-1)} + \frac{c}{2}(\mathbf{x}_j^{(k-1)} - \mathbf{x}_i^{(k-1)}) \quad \forall j \in \mathcal{N}_i, i \in V, \quad (11b)$$

$$\begin{aligned} \mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i} & \phi_i(\mathbf{x}_i) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)})^T \mathbf{x}_i \\ & + c \sum_{j \in \mathcal{N}_i} \left\| \mathbf{x}_i - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right\|_2^2 \quad \forall i \in V, \end{aligned} \quad (11c)$$

where $c > 0$ is a penalty parameter and $\mathbf{u}_{ij}^{(0)} + \mathbf{v}_{ij}^{(0)} = \mathbf{0} \quad \forall i, j$.

Further define $\mathbf{p}_i^{(k)} \triangleq \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)})$, $i \in V$. Then, (11) boils down to the C-ADMM algorithm; see Algorithm 1.

Algorithm 1 C-ADMM for solving (P1)

1: **Given** initial variables $\mathbf{x}_i^{(0)} \in \mathbb{R}^K$ and $\mathbf{p}_i^{(0)} = \mathbf{0}$ for each agent i , $i \in V$. Set $k = 1$.

2: **repeat**

3: For all $i \in V$

4: $\mathbf{p}_i^{(k)} = \mathbf{p}_i^{(k-1)} + c \sum_{j \in \mathcal{N}_i} (\mathbf{x}_i^{(k-1)} - \mathbf{x}_j^{(k-1)})$.

$$\mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i \in \mathcal{X}} f_i(\mathbf{A}_i \mathbf{x}_i) + g_i(\mathbf{x}_i) + \mathbf{x}_i^T \mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} \left\| \mathbf{x}_i - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right\|_2^2. \quad (12)$$

5: **Set** $k = k + 1$.

6: **until** a predefined stopping criterion (e.g., a maximum iteration number) is satisfied.

It is important to note from Step 4 and Step 5 of Algorithm 1 that each agent i updates the variables $(\mathbf{x}_i^{(k)}, \mathbf{p}_i^{(k)})$ in a fully parallel manner, by only using the local function ϕ_i and messages $\{\mathbf{x}_j^{(k-1)}\}_{j \in \mathcal{N}_i}$, which come from its direct neighbors. It has been shown in [18] that, under Assumptions 1 and 2, C-ADMM is guaranteed to converge:

$$\lim_{k \rightarrow \infty} \mathbf{x}_i^{(k)} = \mathbf{x}^*, \quad \lim_{k \rightarrow \infty} (\mathbf{u}_{ij}^{(k)}, \mathbf{v}_{ij}^{(k)}) = (\mathbf{u}_{ij}^*, \mathbf{v}_{ij}^*), \quad \forall j, i, \quad (13)$$

where \mathbf{x}^* and $\{\mathbf{u}_{ij}^*, \mathbf{v}_{ij}^*\}$ denote the optimal primal solution and dual solution to problem (10) (i.e., (P1)), respectively. It is also shown that C-ADMM can converge linearly either when ϕ_i 's are smooth, strongly convex [23] or when ϕ_i 's satisfy certain error bound assumption [22].

One key issue about C-ADMM is that the subproblem in (12) is not always easy to solve. For instance, for the LR function in (6), the associated subproblem (12) is given by

$$\begin{aligned} \mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i \in \mathcal{X}} & \sum_{m=1}^{\bar{M}/N} \log(1 + \exp(-b_{im} \mathbf{a}_{im}^T \mathbf{x}_i)) + g_i(\mathbf{x}_i) \\ & + \mathbf{x}_i^T \mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} \left\| \mathbf{x}_i - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right\|_2^2. \end{aligned} \quad (14)$$

As seen, due to the complicated LR cost, problem (14) cannot yield simple solutions, and a numerical solver has to be employed. Clearly, obtaining a high-accuracy solution of (14) can be computationally expensive, especially when the problem dimension or the number of training data is large. While a low-accuracy solution to (14) can be adopted for complexity reduction, it may destroy the convergence behavior of C-ADMM, as will be shown in Section V.

B. Proposed Inexact C-ADMM

To reduce the complexity of C-ADMM, instead of solving subproblem (12) directly, we consider the following update:

$$\begin{aligned} \mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i \in \mathcal{X}} & \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)})^T \mathbf{A}_i (\mathbf{x}_i - \mathbf{x}_i^{(k-1)}) \\ & + \frac{\beta_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^{(k-1)}\|_2^2 + g_i(\mathbf{x}_i) + \mathbf{x}_i^T \mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} \left\| \mathbf{x}_i - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right\|_2^2, \end{aligned} \quad (15)$$

where $\beta_i > 0$ is a penalty parameter. In (15) we have replaced the smooth cost function $f_i(\mathbf{A}_i \mathbf{x}_i)$ in (12) with its first-order approximation around $\mathbf{x}_i^{(k-1)}$:

$$\nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)})^T \mathbf{A}_i (\mathbf{x}_i - \mathbf{x}_i^{(k-1)}) + \frac{\beta_i}{2} \|\mathbf{x}_i - \mathbf{x}_i^{(k-1)}\|_2^2.$$

To obtain a concise representation of $\mathbf{x}_i^{(k)}$, let us define the *proximity operator* for the non-smooth function g_i at a given point $\mathbf{s} \in \mathbb{R}^K$ as [24]

$$\text{prox}_{g_i}^{\gamma_i}[\mathbf{s}] \triangleq \arg \min_{\mathbf{x} \in \mathcal{X}} g_i(\mathbf{x}) + \frac{\gamma_i}{2} \|\mathbf{x} - \mathbf{s}\|_2^2 \quad (16)$$

where $\gamma_i = \beta_i + 2c|\mathcal{N}_i|$. Clearly, using this definition, (15) is equivalent to the following proximal gradient (PG) step

$$\mathbf{x}_i^{(k)} = \text{prox}_{g_i}^{\gamma_i} \left[\frac{1}{\gamma_i} \left(\beta_i \mathbf{x}_i^{(k-1)} - \mathbf{A}_i^T \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)}) - \mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} (\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}) \right) \right]. \quad (17)$$

PG updates like (17) often admit closed-form expression, especially when g_i 's are sparse promoting functions including the ℓ_1 norm, Euclidean norm, infinity norm and matrix nuclear norm [28]. For example, when $g_i(\mathbf{x}) = \|\mathbf{x}\|_1$ and $\mathcal{X} = \mathbb{R}^K$, (16) has a closed-form solution known as the soft thresholding operator [24], [28]:

$$\mathcal{S} \left[\mathbf{s}, \frac{1}{\gamma_i} \right] = \left(\mathbf{s} - \frac{1}{\gamma_i} \mathbf{1} \right)^+ + \left(-\mathbf{s} - \frac{1}{\gamma_i} \mathbf{1} \right)^+, \quad (18)$$

where $(x)^+ \triangleq \max\{x, 0\}$ and $\mathbf{1}$ is an all-one vector. The IC-ADMM is presented in Algorithm 2.

Algorithm 2 Proposed IC-ADMM for solving (P1)

1: Identical to Algorithm 1 except that (12) is replaced by (17).

Although the idea of “inexact ADMM” is not new, our approach is significantly different from the existing methods [25], [26], where the inexact update is obtained by approximating the quadratic penalization term only. It can be seen that problem (14) is still difficult to solve even the inexact update in [25], [26] is applied. One notable exception is the algorithm proposed in [29], where the cost function is also linearized. However, an additional back substitution step is required, which is not suited for distributed optimization.

The convergence properties of IC-ADMM is characterized by the following theorem.

Theorem 1 Suppose that Assumptions 1, 2(a) and 3 hold. Let

$$\beta_i > \frac{L_{f,i}^2}{\sigma_{f,i}^2} \lambda_{\max}(\mathbf{A}_i^T \mathbf{A}_i) - c \lambda_{\min}(\mathbf{D} + \mathbf{W}) > 0, \quad (19)$$

for all $i \in V$, where λ_{\max} and λ_{\min} denote the maximum and minimum eigenvalues, respectively.

(a) For Algorithm 2, we have $\lim_{k \rightarrow \infty} \mathbf{x}_i^{(k)} = \mathbf{x}^* \forall i \in V$, where \mathbf{x}^* is an optimal solution to (P1).

(b) If $\phi(\mathbf{x})$ is smooth and strongly convex (i.e., g_i 's are removed from (1) and \mathbf{A}_i 's have full column rank) and $\mathcal{X} = \mathbb{R}^K$, then we have

$$\lim_{k \rightarrow \infty} \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_{\frac{1}{2}\mathbf{G} + \alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 = 0 \text{ linearly,}$$

where $\mathbf{x}^{(k)} = [(\mathbf{x}_1^{(k)})^T, \dots, (\mathbf{x}_N^{(k)})^T]^T$; $\mathbf{u}_i^{(k)} \in \mathbb{R}^{K|\mathcal{N}_i|}$ is a vector that stacks $\mathbf{u}_{ij}^{(k)} \forall j \in \mathcal{N}_i$ [see (11a)]; $\mathbf{u}^{(k)} = [(\mathbf{u}_1^{(k)})^T, \dots, (\mathbf{u}_N^{(k)})^T]^T$, and

$$\mathbf{G} \triangleq \mathbf{D}_\beta + c((\mathbf{D} + \mathbf{W}) \otimes \mathbf{I}_K) \succ \mathbf{0}, \quad (20)$$

$$\mathbf{M} \triangleq \tilde{\mathbf{A}}^T (\mathbf{D}_{\sigma_f} - \frac{\rho}{2} \mathbf{I}_{NK}) \tilde{\mathbf{A}} \succ \mathbf{0}, \quad (21)$$

for some $0 < \alpha < 1$ and $\rho > 0$. Here, \otimes denotes the Kronecker product; $\|\mathbf{z}\|_{\mathbf{Z}}^2 \triangleq \mathbf{z}^T \mathbf{A} \mathbf{z}$; \mathbf{I}_K is the $K \times K$ identity matrix; $\tilde{\mathbf{A}} = \text{blkdiag}\{\mathbf{A}_1, \dots, \mathbf{A}_N\}$ (block diagonal); $\mathbf{D}_\beta = \text{diag}\{\beta_1, \dots, \beta_N\} \otimes \mathbf{I}_K$ and $\mathbf{D}_{\sigma_f} = \text{diag}\{\sigma_{f,1}^2, \dots, \sigma_{f,N}^2\} \otimes \mathbf{I}_K$.

The proof is presented in Appendix A. Theorem 1 implies that, given sufficiently large β_i 's, IC-ADMM not only achieves consensus and optimality, but also converges linearly provided that $\phi(\mathbf{x})$ is smooth and strongly convex. It is worth mentioning that our linear convergence analysis generalizes the one presented in [23], which is only true for the C-ADMM.

Remark 1 We remark that the convergence condition in (19) depends on the network topology. Let $\mathbf{L} = \mathbf{D} - \mathbf{W}$ denote the Laplacian matrix of \mathcal{G} . Then $\mathbf{D} + \mathbf{W} = 2\mathbf{D} - \mathbf{L}$. By the graph theory [30], the normalized Laplacian matrix, i.e., $\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$, must have $\lambda_{\max}(\tilde{\mathbf{L}}) \leq 2$. Further, $\lambda_{\max}(\tilde{\mathbf{L}}) < 2$ if and only if the connected graph \mathcal{G} is not bipartite. Thus, we have $\lambda_{\min}(\mathbf{D} + \mathbf{W}) = \lambda_{\min}(\mathbf{D}^{\frac{1}{2}}(2\mathbf{I}_N - \tilde{\mathbf{L}})\mathbf{D}^{\frac{1}{2}}) \geq 0$, and $\lambda_{\min}(\mathbf{D} + \mathbf{W}) > 0$ whenever \mathcal{G} is non-bipartite.

Remark 2 In essence, compared to the C-ADMM in Algorithm 1, the per-iteration computation of IC-ADMM is very simple. As will be presented in Section V, the overall computational complexity of IC-ADMM is lower than that of C-ADMM by an order of magnitude. However, IC-ADMM in general requires more ADMM iterations than C-ADMM to reach the same solution accuracy. This implies that the agents instead have to spend more resources in neighbor-wise communication. We argue that, in some big data applications, computation can be very costly (due to the very large data size) and may even be more expensive than communication since the latter can be made relatively cheaper in certain scenarios. For example, in distributed sparse optimization, the exchanged messages are sparse. Furthermore, for achieving consensus, there is an increasing correlation between exchanged messages; this can be exploited to further reduce the communication rate through simple coding techniques [31].

Besides, communication via wired links (e.g., database servers connected via dedicated fiber links or distributed microprocessors connected by data buses) are much more power/time efficient than wireless links [32]. Therefore, communication are arguably cheaper in these scenarios, and it may be worthy to trade for complexity reduction.

IV. DISTRIBUTED DUAL CONSENSUS ADMM

In this section, we turn the focus to (P2). In Section IV-A, we present a DC-ADMM method for solving (P2). In Section IV-B, an inexact DC-ADMM method is proposed.

A. Proposed DC-ADMM

The DC-ADMM is obtained by applying the C-ADMM (Algorithm 1) to problem (3) which is the Lagrange dual of (P2). Firstly, similar to (10), we write problem (3) as

$$\min_{\substack{\mathbf{y}_1, \dots, \mathbf{y}_N \\ \{\mathbf{t}_{ij}\}}} \sum_{i=1}^N \left(\varphi_i(\mathbf{y}_i) + \frac{1}{N} \mathbf{y}_i^T \mathbf{q} \right) \quad (22a)$$

$$\text{s.t. } \mathbf{y}_i = \mathbf{t}_{ij}, \mathbf{y}_j = \mathbf{t}_{ij} \quad \forall j \in \mathcal{N}_i, i \in V, \quad (22b)$$

where $\mathbf{y}_i \in \mathbb{R}^M$ is the i th agent's local copy of the dual variable \mathbf{y} and φ_i is given in (4). Following a similar argument as in deriving Algorithm 1, we obtain the following update steps at each iteration k

$$\mathbf{u}_{ij}^{(k)} = \mathbf{u}_{ij}^{(k-1)} + \frac{c}{2} (\mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)}) \quad \forall j \in \mathcal{N}_i, i \in V, \quad (23a)$$

$$\mathbf{v}_{ij}^{(k)} = \mathbf{v}_{ij}^{(k-1)} + \frac{c}{2} (\mathbf{y}_j^{(k-1)} - \mathbf{y}_i^{(k-1)}) \quad \forall j \in \mathcal{N}_i, i \in V, \quad (23b)$$

$$\begin{aligned} \mathbf{y}_i^{(k)} = \arg \min_{\mathbf{y}_i \in \mathbb{R}^M} & \varphi_i(\mathbf{y}_i) + \frac{1}{N} \mathbf{y}_i^T \mathbf{q} + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)})^T \mathbf{y}_i \\ & + c \sum_{j \in \mathcal{N}_i} \left\| \mathbf{y}_i - \frac{\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}}{2} \right\|_2^2 \quad \forall i \in V. \end{aligned} \quad (23c)$$

In general, subproblem (23c) is not easy to handle because φ_i is implicit and (23c) is in fact a min-max optimization problem. Fortunately, since (23c) is a strongly convex problem, the Fenchel-Rockafellar duality [33] can be applied so that the min-max problem (23c) reduces to

$$\mathbf{y}_i^{(k)} = \frac{1}{2|\mathcal{N}_i|} \left(\sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) + \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N} \mathbf{q}) - \frac{1}{c} \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) \right), \quad (24)$$

where

$$\begin{aligned} \mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i \in \mathcal{X}_i} & \phi_i(\mathbf{x}_i) + \frac{c}{4|\mathcal{N}_i|} \left\| \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i - \frac{1}{N} \mathbf{q}) \right. \\ & \left. - \frac{1}{c} \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right\|_2^2. \end{aligned} \quad (25)$$

As a result, the min-max subproblem (23c) can actually be obtained by first solving a primal subproblem (25) followed by evaluating $\mathbf{y}_i^{(k)}$ using the close form in (24). By letting $\mathbf{p}_i^{(k)} = \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)})$, $i \in V$, the proposed DC-ADMM is summarized in Algorithm 3.

Algorithm 3 Proposed DC-ADMM for solving (P2)

1: **Given** initial variables $\mathbf{x}_i^{(0)} \in \mathbb{R}^K$, $\mathbf{y}_i^{(0)} \in \mathbb{R}^M$ and $\mathbf{p}_i^{(0)} = \mathbf{0}$ for each agent i , $i \in V$. Set $k = 1$.

2: **repeat**

3: For all $i \in V$

4: $\mathbf{p}_i^{(k)} = \mathbf{p}_i^{(k-1)} + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)})$.

$$\mathbf{x}_i^{(k)} = \arg \min_{\mathbf{x}_i \in \mathcal{X}_i} \phi_i(\mathbf{x}_i) + \frac{c}{4|\mathcal{N}_i|} \left\| \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i - \frac{1}{N} \mathbf{q}) - \frac{1}{c} \mathbf{p}_i^{(k)} + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right\|_2^2. \quad (26)$$

$$\mathbf{y}_i^{(k)} = \frac{1}{2|\mathcal{N}_i|} \left(\sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) - \frac{1}{c} \mathbf{p}_i^{(k)} + \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N} \mathbf{q}) \right). \quad (27)$$

5: **Set** $k = k + 1$.

6: **until** a predefined stopping criterion is satisfied.

We should mention that Algorithm 3 is different from the D-ADMM algorithm in [11, Algorithm 3]. Firstly, Algorithm 3 can be implemented in a fully parallel manner; secondly, Algorithm 3 does not involve solving a min-max problem, thanks to the Fenchel-Rockafellar duality [33].

Interestingly, while DC-ADMM handles the dual problem in (3), it directly yields primal optimal solution of (P2).

Theorem 2 Suppose that Assumptions 1 and 2(b) hold. Then $(\mathbf{y}_1^{(k)}, \dots, \mathbf{y}_N^{(k)})$ converges to a common point \mathbf{y}^* , which is optimal to the dual problem (3). Moreover, any limit point of $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_N^{(k)})$ is primal optimal to (P2).

Proof: Since DC-ADMM is a direct application of C-ADMM to the dual problem (3), it follows from [18] that as $k \rightarrow \infty$,

$$\mathbf{y}_i^{(k)} \rightarrow \mathbf{y}^*, \mathbf{y}_i^{(k)} \rightarrow \mathbf{y}_j^{(k)} \quad \forall j \in \mathcal{N}_i, i \in V. \quad (28)$$

What remains is to show that $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_N^{(k)})$ is asymptotically primal optimal to (P2), i.e., when $k \rightarrow \infty$, the following optimality conditions are satisfied

$$\partial \phi_i(\mathbf{x}_i^{(k)}) + \mathbf{E}_i^T \mathbf{y}^* = \mathbf{0}, \quad i \in V, \quad (29)$$

$$\sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i^{(k)} = \mathbf{q}. \quad (30)$$

To show (29), consider the optimality condition of (25), i.e.,

$$\begin{aligned} \mathbf{0} &= \partial\phi_i(\mathbf{x}_i^{(k)}) + \frac{1}{2|\mathcal{N}_i|} \mathbf{E}_i^T \left(\frac{1}{c} (\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N} \mathbf{q}) \right. \\ &\quad \left. - \frac{1}{c} \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right) \\ &= \partial\phi_i(\mathbf{x}_i^{(k)}) + \mathbf{E}_i^T \mathbf{y}_i^{(k)}, \end{aligned} \quad (31)$$

where the second equality is obtained by (24). Equation (31) infers (29) since $\mathbf{y}_i^{(k)} \rightarrow \mathbf{y}^*$ by (28).

To show (30), rewrite (24) as follows

$$\begin{aligned} \mathbf{0} &= -(\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N} \mathbf{q}) + 2c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} - \frac{\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)}}{2}) \\ &\quad + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)} - \mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)}) \\ &= -(\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N} \mathbf{q}) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} + \mathbf{v}_{ji}^{(k+1)}) + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)} - \mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)}), \end{aligned} \quad (32)$$

where the last equality is obtained by (23a) and (23b). Upon summing (32) over $i = 1, \dots, N$, and by applying (A.11) and (A.12), we can obtain

$$\sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i^{(k)} - \mathbf{q} = c \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)} - \mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)}). \quad (33)$$

Finally, by applying (28) to (33), one obtains (30). \blacksquare

Interestingly, from (33), one observes that the primal feasibility of $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_N^{(k)})$ to (P2) depends on the agents' consensus on the dual variable \mathbf{y} .

B. Proposed Inexact DC-ADMM

In this subsection, we propose an inexact version of DC-ADMM, referred to as the IDC-ADMM. In view of the fact that solving the subproblem in (25) can be expensive, we consider an inexact update of $\mathbf{x}_i^{(k)}$. Specifically, since a non-trivial \mathbf{E}_i can also complicate the solution, we propose to approximate both $f_i(\mathbf{A}_i \mathbf{x}_i)$ and the quadratic term $\frac{c}{4|\mathcal{N}_i|} \left\| \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i - \frac{1}{N} \mathbf{q}) - \frac{1}{c} \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right\|_2^2$ in (25) by their first-order approximations around $\mathbf{x}_i^{(k-1)}$. One can show that this is equivalent to the following update

$$\begin{aligned} \mathbf{x}_i^{(k)} &= \arg \min_{\mathbf{x}_i \in \mathcal{X}_i} (\nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)}))^T \mathbf{A}_i (\mathbf{x}_i - \mathbf{x}_i^{(k-1)}) + g_i(\mathbf{x}_i) \\ &\quad + \frac{1}{2} (\mathbf{x}_i - \mathbf{x}_i^{(k-1)})^T \mathbf{P}_i (\mathbf{x}_i - \mathbf{x}_i^{(k-1)}) + \frac{c}{4|\mathcal{N}_i|} \left\| \frac{1}{c} (\mathbf{E}_i \mathbf{x}_i - \frac{1}{N} \mathbf{q}) \right. \\ &\quad \left. - \frac{1}{c} \mathbf{p}_i^{(k)} + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right\|_2^2, \end{aligned} \quad (34)$$

where²

$$\mathbf{P}_i = \beta_i \mathbf{I}_K - \frac{1}{2c|\mathcal{N}_i|} \mathbf{E}_i^T \mathbf{E}_i.$$

By (16), (34) is equivalent to the following PG update

$$\begin{aligned} \mathbf{x}_i^{(k)} = \text{prox}_{g_i}^{\beta_i} \left[\mathbf{x}_i^{(k-1)} - \frac{1}{\beta_i} \mathbf{A}_i^T \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)}) (\mathbf{E}_i \mathbf{x}_i^{(k-1)} - \frac{1}{N} \mathbf{q}) \right. \\ \left. - \frac{1}{2\beta_i|\mathcal{N}_i|} \mathbf{E}_i^T \left(-\mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right) \right]. \end{aligned} \quad (35)$$

We summarize the proposed IDC-ADMM in Algorithm 4.

Algorithm 4 Proposed IDC-ADMM for solving (P2)

1: Identical to Algorithm 3 except that (26) is replaced by (35).

The convergence property of IDC-ADMM is stated below.

Theorem 3 Suppose that Assumptions 1, 2(b) and 3 hold. Let

$$\mathbf{P}_i - \frac{L_{f,i}^2}{\sigma_{f,i}^2} \mathbf{A}_i^T \mathbf{A}_i \succ \mathbf{0} \quad \forall i \in V. \quad (36)$$

- (a) The sequence $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_N^{(k)})$ generated from Algorithm 4 converges to an optimal solution $\mathbf{x}^* = [(\mathbf{x}_1^{(k)})^T, \dots, (\mathbf{x}_N^{(k)})^T]^T$ to (P2) while $(\mathbf{y}_1^{(k)}, \dots, \mathbf{y}_N^{(k)})$ converges to a common point \mathbf{y}^* which is optimal to problem (3).
- (b) Suppose that each $\phi_i(\mathbf{x}_i)$ is smooth and strongly convex in \mathbf{x}_i , \mathbf{E}_i 's have full row rank and $\mathcal{X}_i = \mathbb{R}^K$ for all $i \in V$. Then, for some $0 < \alpha < 1$ and $\rho > 0$, we have

$$\begin{aligned} \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{\alpha \mathbf{M} + \frac{1}{2} \mathbf{P}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 \\ + \frac{c}{2} \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_{(\mathbf{D} + \mathbf{W}) \otimes \mathbf{I}_M}^2 \rightarrow 0 \text{ linearly,} \end{aligned} \quad (37)$$

where \mathbf{M} is defined in (21), $\tilde{\mathbf{y}}^* \triangleq \mathbf{1}_N \otimes \mathbf{y}^*$ and $\mathbf{P} \triangleq \text{blkdiag}\{\mathbf{P}_1, \dots, \mathbf{P}_N\} \succ \mathbf{0}$.

The proof is presented in Appendix B. Note that, in addition to the smooth and strongly convex objective function, IDC-ADMM also requires matrices \mathbf{E}_i 's to have full row rank in order to have a linear convergence rate.

²When \mathbf{E}_i has orthogonal columns (e.g., $\mathbf{E}_i^T \mathbf{E}_i = \alpha \mathbf{I}$ for some $\alpha \in \mathbb{R}$), then it may not be necessary to approximate the quadratic term. In that case, one instead sets $\mathbf{P}_i = \beta_i \mathbf{I}_K$.

V. NUMERICAL RESULTS

In this section, we examine the numerical performance of Algorithm 1 to 4 presented so far.

A. Performance of C-ADMM and IC-ADMM

To test C-ADMM (Algorithm 1) and IC-ADMM (Algorithm 2), we considered the distributed RPD LR problem in (5) and (6), with $g_i(\mathbf{x}) = \frac{\lambda}{N}\|\mathbf{x}\|_1$ serving as a sparsity promoting function, where $\lambda > 0$ is a penalty parameter. We considered a simple two image classification task. Specifically, we used the images D24 and D68 from the Brodatz data set (<http://www.ux.uis.no/~tranden/brodatz.html>) to generate the regression data matrix \mathbf{A} . We randomly extracted $\bar{M}/2$ overlapping patches with dimension $\sqrt{\bar{K}} \times \sqrt{\bar{K}}$ from the two images, respectively, followed by vectorizing the \bar{M} patches into vectors and stacking all of them into an $\bar{M} \times \bar{K}$ matrix. The rows of the matrix were randomly shuffled and the resultant matrix was used as the data matrix \mathbf{A} . For the RPD LR problem (5), we horizontally partitioned the matrix \mathbf{A} into N submatrices $\mathbf{A}_1, \dots, \mathbf{A}_N$, each with dimension $(\bar{M}/N) \times \bar{K}$. These matrices were used as the training data. Note that each \mathbf{A}_i contains patches from both images. The binary labels \mathbf{b}_i 's then were generated accordingly with 1 for one image and -1 for the other. The feasible set was set to $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^K \mid |x_i| \leq a \ \forall i\}$ for some $a > 0$. The graph \mathcal{G} was randomly generated.

To implement Algorithm 1, we employed the fast iterative shrinkage thresholding algorithm (FISTA) [34], [35] to solve subproblem (12) for each agent i . For (12), the associated FISTA steps can be shown as

$$\tilde{\mathbf{x}}_i^{(\ell)} = \max \left\{ -a, \min \left\{ a, \mathcal{S} \left[\mathbf{z}_i^{(\ell-1)} - \rho_i^{(\ell)} \left[\mathbf{A}_i^T \nabla f_i(\mathbf{A}_i \mathbf{z}_i^{(\ell-1)}) + \mathbf{p}_i^{(k)} + 2c \sum_{j \in \mathcal{N}_i} \left(\mathbf{z}_i^{(\ell-1)} - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right) \right], \frac{\lambda \rho_i^{(\ell)}}{N} \right] \right\} \right\}, \quad (38a)$$

$$\mathbf{z}_i^{(\ell)} = \tilde{\mathbf{x}}_i^{(\ell)} + \frac{\ell-1}{\ell+2}(\tilde{\mathbf{x}}_i^{(\ell)} - \tilde{\mathbf{x}}_i^{(\ell-1)}), \quad (38b)$$

where ℓ denotes the inner iteration index of FISTA, $\rho_i^{(\ell)} > 0$ is a step size and \mathcal{S} is defined in (18). Suppose that FISTA stops at iteration $\ell_i(k)$. We then set $\mathbf{x}_i^{(k)} = \tilde{\mathbf{x}}_i^{(\ell_i(k))}$ as a solution to subproblem (12). The stopping criterion of (38) was based on the PG residue (pgr) $\text{pgr} = \|\mathbf{z}_i^{(\ell-1)} - \tilde{\mathbf{x}}_i^{(\ell)}\|/(\rho_i^{(\ell)}\sqrt{\bar{K}})$ [34], [35]. For obtaining a high-accuracy solution of (12), one may set the stopping criterion as, e.g., $\text{pgr} < 10^{-5}$.

For IC-ADMM, the corresponding step in (17) is given by

$$\mathbf{x}_i^{(k)} = \max \left\{ -a, \min \left\{ a, \frac{1}{\gamma_i} \mathcal{S} \left[\beta \mathbf{x}_i^{(k-1)} - \mathbf{A}_i^T \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k-1)}) - \mathbf{p}_i^{(k)} + c \sum_{j \in \mathcal{N}_i} (\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}), \frac{\lambda}{N} \right] \right\} \right\}. \quad (39)$$

By comparing (39) with (38a), one can see that, for each agent i , the computational complexity of Algorithm 1 per iteration k (we refer this as the “ADMM iteration (ADMM Ite.)”) is roughly $\ell_i(k)$ times that of Algorithm 2. To measure the computational complexity of Algorithm 1, we count the total average number of FISTA iterations implemented by each agent before Algorithm 1 stops. More precisely, suppose that the total number of ADMM iterations of Algorithm 1 is I_{cm} . The complexity per agent due to Algorithm 1 is measured by the computation iteration:

$$\text{Compt. It.} = \frac{1}{N} \sum_{k=1}^{I_{\text{cm}}} \sum_{i=1}^N \ell_i(k). \quad (40)$$

By contrast, the complexity per agent due to Algorithm 2 is simply given by \tilde{I}_{cm} if the total number of ADMM iterations of Algorithm 2 is \tilde{I}_{cm} . The stopping criterion of Algorithms 1 and 2 was based on measuring the solution accuracy $\text{acc} = (\text{obj}(\hat{\mathbf{x}}^{(k)}) - \text{obj}^*) / \text{obj}^*$ and variable consensus error $\text{cserr} = \sum_{i=1}^N \|\hat{\mathbf{x}}^{(k)} - \mathbf{x}_i^{(k)}\|_2^2 / N$, where $\hat{\mathbf{x}}^{(k)} = (\sum_{i=1}^N \mathbf{x}_i^{(k)}) / N$, $\text{obj}(\hat{\mathbf{x}}^{(k)})$ denotes the objective value of (5) given $\mathbf{x} = \hat{\mathbf{x}}^{(k)}$, and obj^* is the optimal value of (5) which was obtained by FISTA [34], [35] with a high solution accuracy of $\text{pgr} < 10^{-6}$. The two algorithms were set to stop whenever acc and cserr are both smaller than preset target values.

In Table I(a), we considered a simulation example of $N = 10$, $\bar{K} = 10,000$, $\bar{M} = 100$, $\lambda = 0.1$, $a = 1$, and display the comparison results. The convergence curves of C-ADMM and IC-ADMM with respect to the ADMM iteration are also shown in Figs. 1(a) and 1(b). The stopping conditions are $\text{acc} < 10^{-4}$ and $\text{cserr} < 10^{-5}$. For C-ADMM, we considered two cases, one with the stopping condition of FISTA for solving subproblem (12) set to $\text{pgr} < 10^{-5}$ and the other with that set to $\text{pgr} < 10^{-4}$. The penalty parameter c for C-ADMM was set to $c = 0.03$ and the step size $\rho_i^{(\ell)}$ of FISTA (see (38)) was set to a constant $\rho_i^{(\ell)} = 0.1$. The penalty parameters c and β of IC-ADMM were set to $c = 0.01$ and $\beta = 1.2$. We observe from Table I(a) that IC-ADMM in general requires more ADMM iterations than C-ADMM; however, the required computation complexity is significantly lower. Specifically, the number of computation iterations of IC-ADMM is around $81,459/2973 \approx 27.4$ times smaller than that of C-ADMM ($\text{pgr} < 10^{-5}$). We also observe that C-ADMM ($\text{pgr} < 10^{-4}$) consumes a smaller number of computation iterations for achieving $\text{acc} < 10^{-4}$. However, the associated $\text{cserr} = 3.425 \times 10^{-4}$ is quite large. In fact, C-ADMM ($\text{pgr} < 10^{-4}$) cannot reduce cserr properly. As one can see from Fig. 1(b), the

TABLE I: Comparison of C-ADMM and IC-ADMM

(a) $N = 10, \bar{K} = 10,000, \bar{M} = 100, \lambda = 0.1, a = 1.$

	C-ADMM (pgr < 10^{-5})	C-ADMM (pgr < 10^{-4})	IC-ADMM
ADMM lte.	810	675	2973
Compt. lte.	81,459	30,648	2973
acc < 10^{-4}	9.982×10^{-5}	9.91×10^{-5}	9.99×10^{-5}
cserr < 10^{-5}	1.53×10^{-6}	3.425×10^{-4}	3.859×10^{-9}

(b) $N = 50, \bar{K} = 10,000, \bar{M} = 500, \lambda = 0.15, a = 1.$

	C-ADMM (pgr < 10^{-5})	C-ADMM (pgr < 10^{-4})	IC-ADMM
ADMM lte.	952	N/A	7,251
Compt. lte.	1.432×10^5	N/A	7,251
acc < 10^{-4}	9.99×10^{-5}	N/A	9.999×10^{-5}
cserr < 10^{-5}	1.305×10^{-7}	N/A	1.169×10^{-10}

cserr curve of C-ADMM (pgr < 10^{-4}) keeps relatively high and does not decrease along the iterations. When one further reduces the accuracy of FISTA to pgr < 10^{-3} , C-ADMM converges very slowly, as shown in Figs. 1(a) and 1(b). We also plot the convergence curves of the consensus subgradient method in [6], where the diminishing step size $10/k$ was used. As one can see, the consensus subgradient method converges much slower than IC-ADMM.

In Table I(b), we considered another example with the network size increased to $N = 50$. We set $c = 0.004$ for C-ADMM and $\rho_i^{(\ell)} = 0.1$ for FISTA; while for IC-ADMM, we set $c = 0.008$ and $\beta = 1.2$. We can observe similar comparison results from Table I(b). Specifically, the number of computation iterations of IC-ADMM is around 19.7 times smaller than C-ADMM (pgr < 10^{-5}). When considering a lower accuracy of pgr < 10^{-4} , it is found that C-ADMM cannot properly converge.

In contrast to Fig. 1(a), Fig. 1(c) displays the convergence curves of C-ADMM (pgr < 10^{-5}) and IC-ADMM with respect to the computation iteration. Unlike Fig. 1(a), here each ADMM iteration k is expanded to $\frac{1}{N} \sum_{i=1}^N \ell_i(k)$ computation iterations (see the zoom-ined box in Fig. 1(c)). Interestingly, as one can see from this figure, IC-ADMM converges much faster than C-ADMM from this complexity point of view.

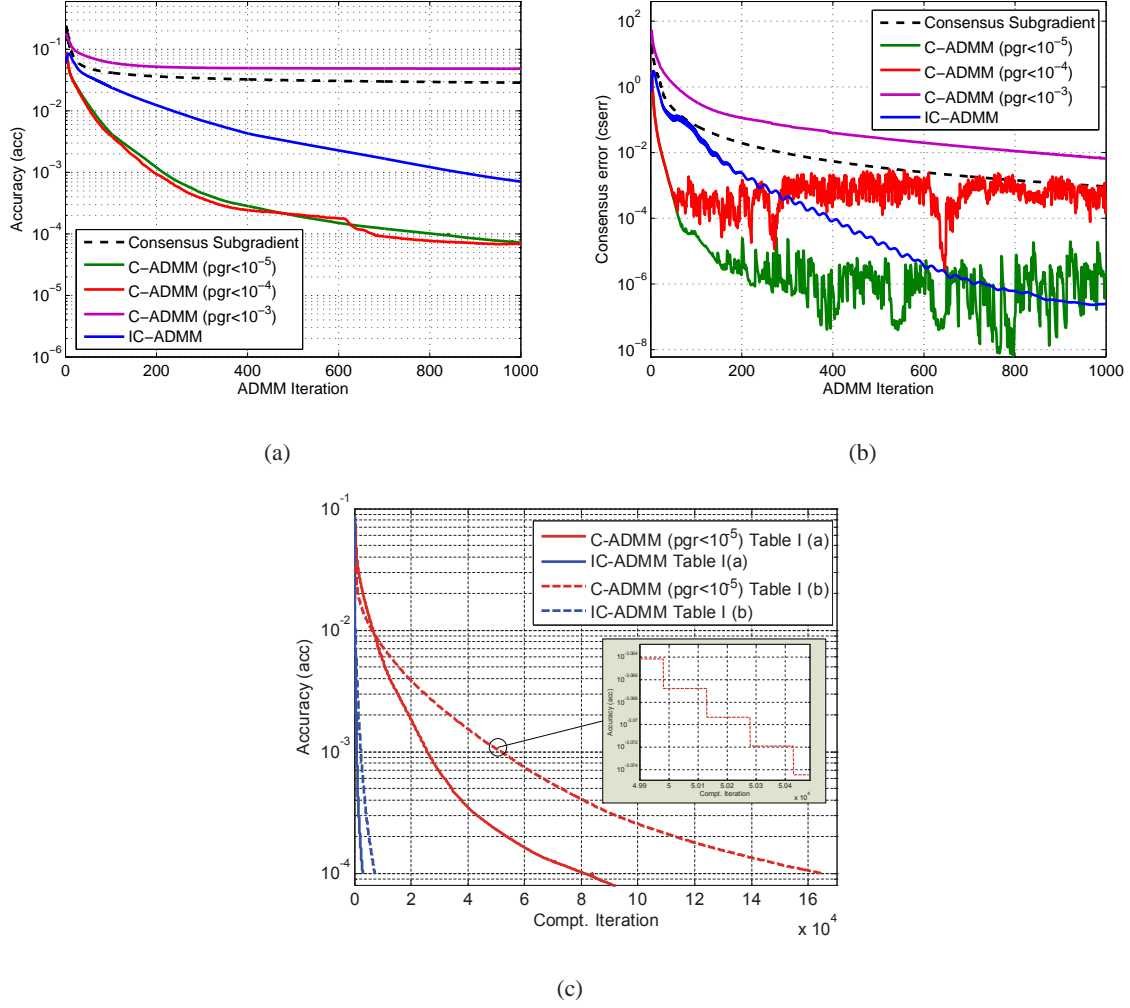


Fig. 1: Convergence curves of C-ADMM and IC-ADMM.

B. Performance of DC-ADMM and IDC-ADMM

We examine the performance of DC-ADMM (Algorithm 3) and IDC-ADMM (Algorithm 4) by considering the distributed CPD LR problem in (7), with $\Psi_i(\mathbf{x}_i; \mathbf{E}_i, \mathbf{b}) = \sum_{m=1}^{\bar{M}} \log(1 + \exp(-b_m \sum_{i=1}^N \mathbf{e}_{im}^T \mathbf{x}_i)) + \frac{\lambda}{N} \|\mathbf{x}_i\|_1$ (i.e., $g_i(\mathbf{x}_i) = \frac{\lambda}{N} \|\mathbf{x}_i\|_1$) and $\mathcal{X}_i = \{\mathbf{x}_i \in \mathbb{R}^{\bar{K}/N} \mid |[\mathbf{x}_i]_j| \leq a \ \forall j\}$ for some $a > 0$. DC-ADMM and IDC-ADMM were applied to handle the associated problem (8). The regression data matrix $\mathbf{A} = [\mathbf{E}_1, \dots, \mathbf{E}_N]$ was generated following the same way as in Section V-A. To implement DC-ADMM, we employed FISTA [34], [35] to solve the subproblem (26). The involved steps are similar to those in (38) for C-ADMM, and the solution accuracy of subproblem (26) was measured by the PG residue pgr of FISTA. We also counted the computation complexity of DC-ADMM by the same way as in (40).

Similarly, the stopping criterion of Algorithms 3 and 4 was based on measuring the solution accuracy $\text{acc} = (\text{obj}(\mathbf{x}^{(k)}) - \text{obj}^*)/\text{obj}^*$, where $\mathbf{x}^{(k)} = [(\mathbf{x}_1^{(k)})^T, \dots, (\mathbf{x}_N^{(k)})^T]^T$, $\text{obj}(\mathbf{x}^{(k)})$ denotes the objective value of (7) and obj^* is the optimal value of (7) obtained by using FISTA with a high accuracy of $\text{pgr} < 1e^{-6}$. Unlike Section V-A, here for problem (7) we did not care the consensus of $\{\mathbf{y}_i^{(k)}\}$ or the satisfaction of constraint $\sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i^{(k)} = \mathbf{z}^{(k)}$ since $\mathbf{x}^{(k)}$ is always feasible and is an approximate solution to the original problem (7).

In Table II(a), we show the comparison results for an example of $N = 50$, $\bar{K} = 10,000$, $\bar{M} = 100$, $\lambda = 0.05$ and $a = 10$. The convergence curves are also shown in Figs. 2(a) to 2(c). It was set $c = 0.05$ for DC-ADMM and the step size of FISTA $\rho_i^{(\ell)}$ was determined based on a line search rule [35]. We see from Table II(a) that, for achieving $\text{acc} < 10^{-4}$, DC-ADMM ($\text{pgr} < 10^{-5}$) took 329 ADMM iterations whereas IDC-ADMM took 10,814 iterations. However, the computation complexity of DC-ADMM ($\text{pgr} < 10^{-5}$) is around $3.711 \times 10^5 / 10814 \approx 29$ times higher than IDC-ADMM. When one reduce the solution accuracy of FISTA for solving subproblem (26) to $\text{pgr} < 10^{-4}$, DC-ADMM cannot reach the high accuracy of $\text{acc} < 10^{-4}$, as observed in Fig. 2(a). From Fig. 2(b), one can see that DC-ADMM converges much faster than IDC-ADMM with respect to the ADMM iterations. However, as shown from Fig. 2(c), the comparison result is reversed when one counts the computation iterations.

In Table II(b), we considered another example with \bar{K} increased to 40,000. We set $c = 0.05$ for DC-ADMM, and set $c = 0.08$ and $\beta = 5$ for IDC-ADMM. From Table II(b) and Figs. 2(b) and 2(c), one can observe similar results.

Remark 3 From the simulation results, we have seen that IC-ADMM and IDC-ADMM can respectively gain significant complexity reduction compared to C-ADMM and DC-ADMM, but require more ADMM iterations, i.e., neighbor-wise communications. To further illustrate this aspect, let us assume that an agent consumes resource R_{cm} (e.g., power or time) for communication and R_{cp} for computation. Then IC-ADMM (IDC-ADMM) would be more efficient than C-ADMM (DC-ADMM) if

$$\begin{aligned} \tilde{I}_{\text{cm}}(R_{\text{cm}} + R_{\text{cp}}) &\leq I_{\text{cm}}R_{\text{cm}} + I_{\text{cp}}R_{\text{cp}} \\ \Leftrightarrow R_{\text{cm}} &\leq R_{\text{cp}} \frac{I_{\text{cp}} - \tilde{I}_{\text{cm}}}{\tilde{I}_{\text{cm}} - I_{\text{cm}}}, \end{aligned} \quad (41)$$

where \tilde{I}_{cm} is the number of ADMM iterations of IC-ADMM (IDC-ADMM) and I_{cm} and I_{cp} are the numbers of ADMM and computation iterations of C-ADMM (DC-ADMM). By taking Table II(b) as the example, (41) indicates that the inexact consensus methods are more efficient if $R_{\text{cm}} \leq 45.65R_{\text{cp}}$. As we discussed in Remark 2, R_{cp} could be large for large-scale problems whereas R_{cm} may be made cheaper in some wired scenarios by exploiting variable sparsity and correlations.

TABLE II: Comparison of DC-ADMM and IDC-ADMM

(a) $N = 50, \bar{K} = 10,000, \bar{M} = 100, \lambda = 0.05, a = 10.$

	DC-ADMM (pgr $< 10^{-5}$)	DC-ADMM (pgr $< 10^{-4}$)	IDC-ADMM
ADMM lte.	329	N/A	10814
Compt. lte.	3.711×10^5	N/A	10814
acc $< 10^{-4}$	9.928×10^{-5}	N/A	9.997×10^{-4}

(b) $N = 50, \bar{K} = 40,000, \bar{M} = 100, \lambda = 0.01, a = 20.$

	DC-ADMM (pgr $< 10^{-5}$)	DC-ADMM (pgr $< 10^{-4}$)	IDC-ADMM
ADMM lte.	475	N/A	38728
Compt. lte.	1.768×10^6	N/A	38728
acc $< 10^{-4}$	9.777×10^{-4}	N/A	9.999×10^{-5}

VI. CONCLUSIONS

In this paper, we have presented ADMM based distributed optimization methods for solving problems (P1) and (P2) in multi-agent networks. In particular, aiming at reducing the computational complexity of C-ADMM for solving large-scale instances of (P1) with complicated objective functions, we have proposed the IC-ADMM method (Algorithm 2) where agents perform one PG update only at each iteration. For (P2), we have proposed the DC-ADMM method (Algorithm 3) and its complexity reduced counterpart IDC-ADMM (Algorithm 4). Preliminary numerical results based on the distributed LR problems (5) and (8) have shown that both IC-ADMM and IDC-ADMM require more ADMM iterations than C-ADMM and DC-ADMM, but the traded computational complexity reduction is quite significant, demonstrating potential for distributed sparse big data applications.

APPENDIX A

PROOF OF THEOREM 1

Proof of Theorem 1(a): Note that, without loss generality, we can assume that $\mathbf{x}_i \in \mathbb{R}^K \forall i$ in (P1), problem (10) and subproblem (15), because \mathcal{X} can be implicitly represented by an indicator function and included in the nonsmooth component g_i 's; see [36, Section 5]. Let $(\mathbf{x}_1^*, \dots, \mathbf{x}_N^*)$ and $\{\mathbf{u}_{ij}^*, \mathbf{v}_{ij}^*, j \in \mathcal{N}_i\}_{i=1}^N$ be a pair of optimal primal and dual solutions to problem (10). Then they satisfy the following

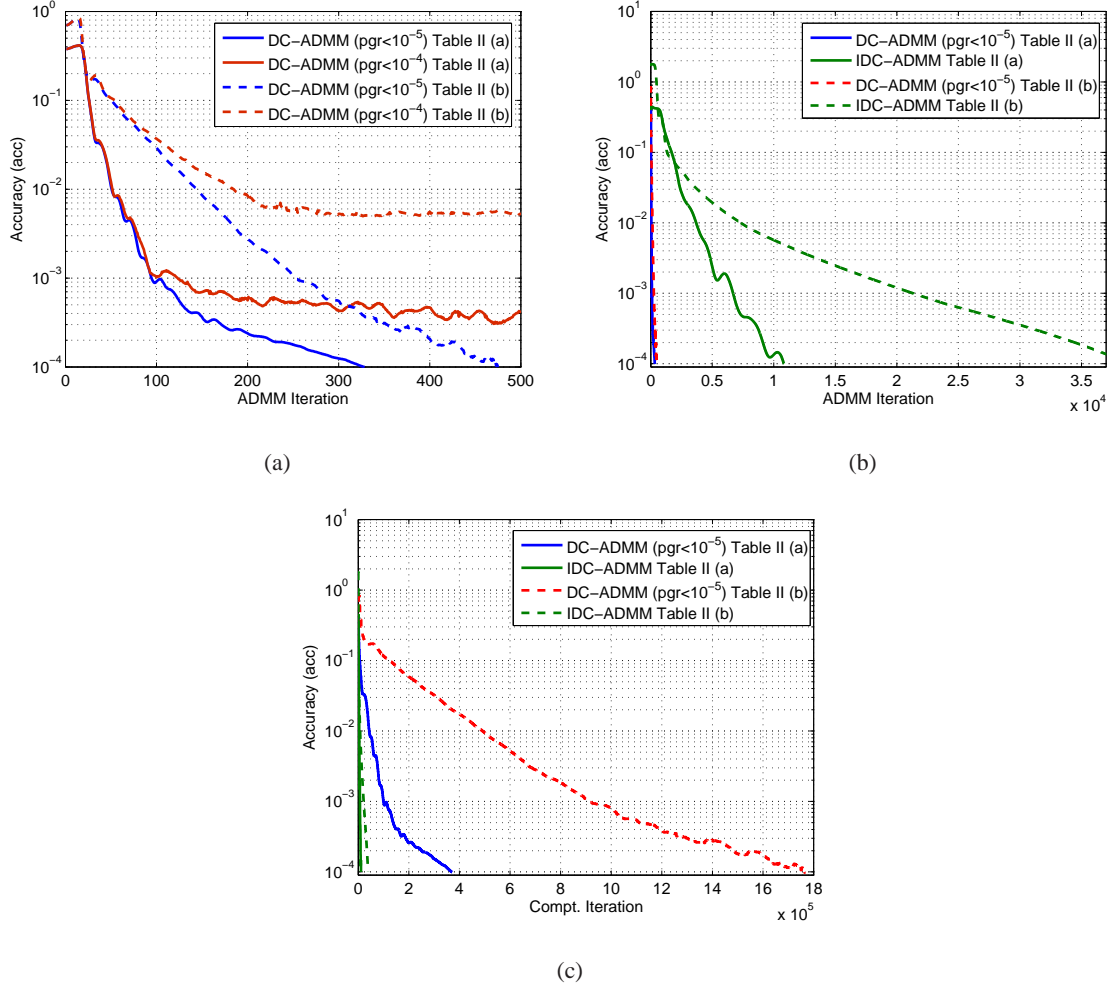


Fig. 2: Convergence curves of DC-ADMM and IDC-ADMM.

Karush-Kuhn-Tucker (KKT) conditions: $\forall i \in V$,

$$\mathbf{A}_i^T \nabla f_i(\mathbf{A}_i \mathbf{x}_i^*) + \partial g_i(\mathbf{x}_i^*) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^* + \mathbf{v}_{ji}^*) = \mathbf{0}, \quad (\text{A.1})$$

$$\mathbf{x}_i^* = \mathbf{x}_j^* \quad \forall j \in \mathcal{N}_i, \quad (\text{A.2})$$

$$\mathbf{u}_{ij}^* + \mathbf{v}_{ij}^* = \mathbf{0}, \quad \forall j \in \mathcal{N}_i. \quad (\text{A.3})$$

Under Assumption 1, (A.2) implies that $\mathbf{x}^* \triangleq \mathbf{x}_1^* = \dots = \mathbf{x}_N^*$, i.e., consensus among agents is reached, and thus \mathbf{x}^* is optimal to the original problem (P1).

Let $\mathbf{s}_i^{(k)} = \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k)})$ and $\mathbf{s}_i^* = \nabla f_i(\mathbf{A}_i \mathbf{x}_i^*)$. By recalling that $\mathbf{p}_i^{(k)} = \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)})$, $i \in V$,

and by the optimality condition of (15) [14] and (A.1), we have that

$$\begin{aligned}
& \mathbf{A}_i^T \mathbf{s}_i^{(k-1)} + \beta_i(\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) + \partial g_i(\mathbf{x}_i^{(k)}) \\
& + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + 2c \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i^{(k)} - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right) \\
& = \mathbf{0} = \mathbf{A}_i^T \mathbf{s}_i^* + \partial g_i(\mathbf{x}_i^*) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^* + \mathbf{v}_{ji}^*).
\end{aligned} \tag{A.4}$$

Adding and subtracting $\mathbf{A}_i^T \mathbf{s}_i^{(k)}$ in the left hand side (LHS) of (A.4) followed by multiplying $(\mathbf{x}_i^{(k)} - \mathbf{x}^*)$ on both sides yields

$$\begin{aligned}
& (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^{(k)})^T \mathbf{A}_i(\mathbf{x}_i^{(k)} - \mathbf{x}^*) + \beta_i(\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) + (\mathbf{s}_i^{(k)} - \mathbf{s}_i^*)^T \mathbf{A}_i(\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\
& + (\partial g_i(\mathbf{x}_i^{(k)}) - \partial g_i(\mathbf{x}^*))^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} - \mathbf{u}_{ij}^* + \mathbf{v}_{ji}^{(k)} - \mathbf{v}_{ji}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\
& + 2c \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i^{(k)} - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right)^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) = \mathbf{0}.
\end{aligned} \tag{A.5}$$

Note that the first term on the LHS of (A.5) can be lower bounded as

$$\begin{aligned}
& (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^{(k)})^T \mathbf{A}_i(\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\
& \geq \frac{-1}{2\rho} \|\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^{(k)}\|_2^2 - \frac{\rho}{2} \|\mathbf{x}_i^{(k)} - \mathbf{x}^*\|_{\mathbf{A}_i^T \mathbf{A}_i}^2 \\
& \geq \frac{-L_{f,i}^2}{2\rho} \|\mathbf{x}_i^{(k-1)} - \mathbf{x}_i^{(k)}\|_{\mathbf{A}_i^T \mathbf{A}_i}^2 - \frac{\rho}{2} \|\mathbf{x}_i^{(k)} - \mathbf{x}^*\|_{\mathbf{A}_i^T \mathbf{A}_i}^2
\end{aligned} \tag{A.6}$$

for any $\rho > 0$, where the second inequality is due to (9) in Assumption 3. By the strong convexity of f_i and convexity of g_i , the third and fourth terms of (A.5) can respectively be lower bounded as

$$(\mathbf{s}_i^{(k)} - \mathbf{s}_i^*)^T \mathbf{A}_i(\mathbf{x}_i^{(k)} - \mathbf{x}^*) \geq \sigma_{f,i}^2 \|\mathbf{x}_i^{(k)} - \mathbf{x}^*\|_{\mathbf{A}_i^T \mathbf{A}_i}^2, \tag{A.7}$$

$$(\partial g_i(\mathbf{x}_i^{(k)}) - \partial g_i(\mathbf{x}^*))^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \geq 0. \tag{A.8}$$

Moreover, it follows from (11a) and (11b) that the fifth term of (A.5) can be expressed as

$$\begin{aligned}
& \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} - \mathbf{u}_{ij}^* + \mathbf{v}_{ji}^{(k)} - \mathbf{v}_{ji}^*) = \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^* \\
& + \mathbf{v}_{ji}^{(k+1)} - \mathbf{v}_{ji}^*) - 2c \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i^{(k)} - \frac{\mathbf{x}_i^{(k)} + \mathbf{x}_j^{(k)}}{2} \right).
\end{aligned} \tag{A.9}$$

By substituting (A.6) to (A.9) into (A.5) and summing over $i = 1, \dots, N$, we obtain

$$\begin{aligned}
& \|\mathbf{x}^{(k)} - \mathbf{x}^*\|_{\mathbf{M}}^2 - \frac{1}{2\rho} \|\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}\|_{\tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2 + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})^T \mathbf{D}_\beta (\mathbf{x}^{(k)} - \mathbf{x}^*) \\
& + \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}_i^*) + \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{v}_{ji}^{(k+1)} - \mathbf{v}_{ji}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}_i^*) \\
& + 2c \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \left(\frac{\mathbf{x}_i^{(k)} + \mathbf{x}_j^{(k)}}{2} - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right)^T (\mathbf{x}_i^{(k)} - \mathbf{x}_i^*) \leq 0,
\end{aligned} \tag{A.10}$$

where $\mathbf{x}^{(k)} = [(\mathbf{x}_1^{(k)})^T, \dots, (\mathbf{x}_N^{(k)})^T]^T$, $\tilde{\mathbf{A}} = \text{blkdiag}\{\mathbf{A}_1, \dots, \mathbf{A}_N\}$, $\mathbf{D}_{L_f} = \text{diag}\{L_{f,1}^2, \dots, L_{f,N}^2\} \otimes \mathbf{I}_K$, $\mathbf{D}_\beta = \text{diag}\{\beta_1, \dots, \beta_N\} \otimes \mathbf{I}_K$ and \mathbf{M} is defined in (21).

It can be observed from (11a) and (11b) that

$$\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ij}^{(k)} = \mathbf{0} \quad \forall j, i, k, \quad (\text{A.11})$$

given the initial $\mathbf{u}_{ij}^{(0)} + \mathbf{v}_{ij}^{(0)} = \mathbf{0} \quad \forall j, i, k$. Besides, due to the symmetric property of \mathbf{W} , for any $\{\alpha_{ij}\}$, we have

$$\begin{aligned} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \alpha_{ij} &= \sum_{i=1}^N \sum_{j=1}^N [\mathbf{W}]_{i,j} \alpha_{ij} \\ &= \sum_{i=1}^N \sum_{j=1}^N [\mathbf{W}]_{i,j} \alpha_{ji} = \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \alpha_{ji}. \end{aligned} \quad (\text{A.12})$$

By the above two properties and the fact of $\mathbf{u}_{ij}^* + \mathbf{v}_{ij}^* = \mathbf{0} \quad \forall j \in \mathcal{N}_i$ [Eqn. (A.3)], the fourth and fifth terms in the LHS of (A.10) can be written as

$$\begin{aligned} &\sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &\quad + \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{v}_{ji}^{(k+1)} - \mathbf{v}_{ji}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &= \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^*)^T (\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}) \\ &= \frac{2}{c} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^*)^T (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^{(k)}) \\ &\triangleq \frac{2}{c} (\mathbf{u}^{(k+1)} - \mathbf{u}^*)^T (\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}), \end{aligned} \quad (\text{A.13})$$

where the second equality is owing to (11a), and $\mathbf{u}^{(k)}$ is a vector that stacks $\mathbf{u}_{ij}^{(k)}$ for all $j \in \mathcal{N}_i$, $i = 1, \dots, N$. The sixth term in the LHS of (A.10) can be rearranged as follows

$$\begin{aligned} &c \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &\quad + c \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j^{(k)} - \mathbf{x}_j^{(k-1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &= c \sum_{i=1}^N |\mathcal{N}_i| (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &\quad + c \sum_{i=1}^N \sum_{j=1}^N [\mathbf{W}]_{i,j} (\mathbf{x}_j^{(k)} - \mathbf{x}_j^{(k-1)})^T (\mathbf{x}_i^{(k)} - \mathbf{x}^*) \\ &= c (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})^T [(\mathbf{D} + \mathbf{W}) \otimes \mathbf{I}_K] (\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*), \end{aligned} \quad (\text{A.14})$$

where $\tilde{\mathbf{x}}^* = \mathbf{1}_N \otimes \mathbf{x}^*$. Let $\mathbf{L} = \mathbf{D} - \mathbf{W}$ be the Laplacian matrix of \mathcal{G} . By the graph theory [30], the normalized Laplacian matrix, i.e., $\tilde{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$, have $\lambda_{\max}(\tilde{\mathbf{L}}) \leq 2$. Thus, in (A.14),

$$\mathbf{D} + \mathbf{W} = 2\mathbf{D} - \mathbf{L} = \mathbf{D}^{\frac{1}{2}} (2\mathbf{I}_N - \tilde{\mathbf{L}}) \mathbf{D}^{\frac{1}{2}} \succeq \mathbf{0}. \quad (\text{A.15})$$

By substituting (A.13) and (A.14) into (A.10) and by applying the fact of

$$\begin{aligned} (\mathbf{a}^{(k)} - \mathbf{a}^{(k-1)})^T \mathbf{Q} (\mathbf{a}^{(k)} - \mathbf{a}^*) &= \frac{1}{2} \|\mathbf{a}^{(k)} - \mathbf{a}^*\|_{\mathbf{Q}}^2 \\ &+ \frac{1}{2} \|\mathbf{a}^{(k)} - \mathbf{a}^{(k-1)}\|_{\mathbf{Q}}^2 - \frac{1}{2} \|\mathbf{a}^{(k-1)} - \mathbf{a}^*\|_{\mathbf{Q}}^2 \end{aligned} \quad (\text{A.16})$$

for any sequence $\mathbf{a}^{(k)}$ and matrix \mathbf{Q} , one obtains that

$$\begin{aligned} &(\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*)^T \left[\mathbf{M} + \frac{1}{2} \mathbf{G} \right] (\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*) + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 \\ &\leq \frac{1}{2} (\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*)^T \mathbf{G} (\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*) \\ &\quad + \frac{1}{c} \|\mathbf{u}^{(k)} - \mathbf{u}^*\|_2^2 - \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 \\ &\quad - (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})^T \left[\frac{1}{2} \mathbf{G} - \frac{1}{2\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \right] (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}), \end{aligned} \quad (\text{A.17})$$

where $\mathbf{G} \triangleq \mathbf{D}_\beta + c((\mathbf{D} + \mathbf{W}) \otimes \mathbf{I}_K) \succ \mathbf{0}$ as defined in (20). It thus follows from (A.17) that, to ensure

$$\mathbf{x}^{(k)} \rightarrow \tilde{\mathbf{x}}^*, \quad \mathbf{u}^{(k)} \rightarrow \mathbf{u}^*, \quad (\text{A.18a})$$

$$\mathbf{x}^{(k)} \rightarrow \mathbf{x}^{(k-1)}, \quad \mathbf{u}^{(k+1)} \rightarrow \mathbf{u}^{(k)}, \quad (\text{A.18b})$$

as $k \rightarrow \infty$, it suffices to have $\mathbf{M} \succeq \mathbf{0}$ and $\mathbf{G} - \frac{1}{\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \succeq \mathbf{0}$, which can be achieved by

$$\sigma_{f,i}^2 - \frac{\rho}{2} > 0 \quad (\text{A.19a})$$

$$\beta_i \mathbf{I}_K + c \lambda_{\min}(\mathbf{D} + \mathbf{W}) \mathbf{I}_k - \frac{L_{f,i}^2}{\rho} \mathbf{A}_i^T \mathbf{A}_i \succ \mathbf{0}, \quad (\text{A.19b})$$

for all $i \in V$. One can verify that β_i in (19) guarantees (A.19) to hold for some $\rho > 0$. Finally, by (A.11) and by applying (A.18) to (A.4), we conclude that the optimality conditions in (A.1), (A.2) and (A.3) are satisfied as $k \rightarrow \infty$. The proof is thus complete. \blacksquare

Proof of Theorem 1(b): Let $0 < \alpha < 1$ be some positive number and rewrite (A.17) as

$$\begin{aligned} &\left(\|\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*\|_{\frac{1}{2}\mathbf{G} + \alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 \right) + \|\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*\|_{(1-\alpha)\mathbf{M}}^2 \\ &\quad + \|\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*\|_{\alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 \\ &\quad + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})^T \left[\frac{1}{2} \mathbf{G} - \frac{1}{2\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \right] (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) \\ &\leq \left(\|\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*\|_{\frac{1}{2}\mathbf{G} + \alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k)} - \mathbf{u}^*\|_2^2 \right). \end{aligned} \quad (\text{A.20})$$

Then, it is sufficient to show that, for some $\delta > 0$,

$$\begin{aligned}
& \|\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*\|_{(1-\alpha)\mathbf{M}}^2 + \|\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*\|_{\alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 \\
& + (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})^T \left[\frac{1}{2} \mathbf{G} - \frac{1}{2\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \right] (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) \\
& \geq \delta \left(\|\mathbf{x}^{(k)} - \tilde{\mathbf{x}}^*\|_{\frac{1}{2}\mathbf{G} + \alpha\mathbf{M}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 \right).
\end{aligned} \tag{A.21}$$

Recall from (A.4) and (A.9) that

$$\begin{aligned}
& \mathbf{A}_i^T \mathbf{s}_i^{(k-1)} - \mathbf{A}_i^T \mathbf{s}_i^* + \beta_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^*) \\
& + \sum_{j \in \mathcal{N}_i} (\mathbf{v}_{ji}^{(k+1)} - \mathbf{v}_{ji}^*) + 2c \sum_{j \in \mathcal{N}_i} \left(\frac{\mathbf{x}_i^{(k)} + \mathbf{x}_j^{(k)}}{2} - \frac{\mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k-1)}}{2} \right) \\
& = \mathbf{0}.
\end{aligned} \tag{A.22}$$

By applying (A.3) and (A.11), (A.22) can be expressed as

$$\begin{aligned}
& \mathbf{A}_i^T \mathbf{s}_i^{(k-1)} - \mathbf{A}_i^T \mathbf{s}_i^* + \beta_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} - \mathbf{u}_{ij}^{(k+1)}) \\
& - \mathbf{u}_{ij}^* + \mathbf{u}_{ji}^* + c \sum_{j \in \mathcal{N}_i} \left(\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)} + \mathbf{x}_j^{(k)} - \mathbf{x}_j^{(k-1)} \right) = \mathbf{0}.
\end{aligned} \tag{A.23}$$

After stacking (A.23) for $i = 1, \dots, N$, one obtains

$$\begin{aligned}
& \tilde{\mathbf{A}}^T (\mathbf{s}^{(k-1)} - \mathbf{s}^*) + \mathbf{G} (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}) \\
& + \boldsymbol{\Psi} (\mathbf{u}^{(k+1)} - \mathbf{u}^*) = \mathbf{0}.
\end{aligned} \tag{A.24}$$

where $\mathbf{s}^{(k)} = ((\mathbf{s}_1^{(k)})^T, \dots, (\mathbf{s}_N^{(k)})^T)^T$ and $\boldsymbol{\Psi} \in \mathbb{R}^{KN \times 2|\mathcal{E}|K}$ is a linear mapping matrix satisfying

$$\begin{bmatrix} \sum_{j \in \mathcal{N}_1} (\mathbf{u}_{1j}^{(k+1)} - \mathbf{u}_{j1}^{(k+1)}) \\ \vdots \\ \sum_{j \in \mathcal{N}_N} (\mathbf{u}_{Nj}^{(k+1)} - \mathbf{u}_{jN}^{(k+1)}) \end{bmatrix} = \boldsymbol{\Psi} \mathbf{u}^{(k+1)}. \tag{A.25}$$

According to [23], both $\mathbf{u}^{(k+1)}$ and \mathbf{u}^* lie in the range space of $\boldsymbol{\Psi}^T$. Hence, one can show that

$$\|\boldsymbol{\Psi} (\mathbf{u}^{(k+1)} - \mathbf{u}^*)\|^2 \geq \sigma_{\min}^2(\boldsymbol{\Psi}) \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 \tag{A.26}$$

where $\sigma_{\min}(\boldsymbol{\Psi}) > 0$ is the minimum nonzero singular value of $\boldsymbol{\Psi}$. From (A.24), we have the following chain

$$\begin{aligned}
& \|\mathbf{G} (\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})\|^2 = \|\tilde{\mathbf{A}}^T (\mathbf{s}^{(k-1)} - \mathbf{s}^*) - \boldsymbol{\Psi} (\mathbf{u}^{(k+1)} - \mathbf{u}^*)\|^2 \\
& \geq (1 - \mu) \|\tilde{\mathbf{A}}^T (\mathbf{s}^{(k-1)} - \mathbf{s}^*)\|^2 + (1 - \frac{1}{\mu}) \|\boldsymbol{\Psi} (\mathbf{u}^{(k+1)} - \mathbf{u}^*)\|^2
\end{aligned}$$

$$\begin{aligned}
&\geq (1 - \mu)\lambda_{\max}(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}}) \|(\mathbf{s}^{(k-1)} - \mathbf{s}^*)\|^2 + (1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi) \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|^2 \\
&\geq (1 - \mu)\lambda_{\max}(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}}) \|(\mathbf{x}^{(k-1)} - \mathbf{x}^*)\|_{\tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2 + (1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi) \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|^2, \tag{A.27}
\end{aligned}$$

where the first inequality is due to the fact that

$$\|\mathbf{a} + \mathbf{q}\|_2^2 \geq (1 - \mu)\|\mathbf{a}\|_2^2 + (1 - \frac{1}{\mu})\|\mathbf{q}\|_2^2 \tag{A.28}$$

for any \mathbf{a}, \mathbf{q} and $\mu > 0$, the second inequality is obtained by setting $\mu > 1$ and (A.26), and the last inequality is by (9). Equation (A.27) implies that

$$\begin{aligned}
\frac{\delta}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 &\leq \frac{\delta}{c(1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi)} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_{\mathbf{G}^T \mathbf{G}}^2 \\
&+ \frac{\delta(\mu - 1)\lambda_{\max}(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}})}{c(1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi)} \|(\mathbf{x}^{(k-1)} - \tilde{\mathbf{x}}^*)\|_{\tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2. \tag{A.29}
\end{aligned}$$

According to (A.29), (A.21) can hold true for some $\delta > 0$ if the following three conditions can be satisfied

$$(1 - \alpha)\mathbf{M} \succeq \delta \left(\frac{1}{2}\mathbf{G} + \alpha\mathbf{M} \right), \tag{A.30a}$$

$$\alpha(\mathbf{D}_{\sigma_f} - \frac{\rho}{2}\mathbf{I}_{NK}) \succeq \delta \frac{(\mu - 1)\lambda_{\max}(\tilde{\mathbf{A}}^T \tilde{\mathbf{A}})}{c(1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi)} \mathbf{D}_{L_f}, \tag{A.30b}$$

$$\frac{1}{2}\mathbf{G} - \frac{1}{2\rho}\tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \succeq \frac{\delta}{c(1 - \frac{1}{\mu})\sigma_{\min}^2(\Psi)} \mathbf{G}^T \mathbf{G}. \tag{A.30c}$$

Note that, given β_i 's in (19) and full column rank \mathbf{A}_i 's, (A.19) holds and, moreover, $\mathbf{M} \succ \mathbf{0}$, $\mathbf{D}_{\sigma_f} - \frac{\rho}{2}\mathbf{I}_{NK} \succ \mathbf{0}$ and $\mathbf{G} - \frac{1}{\rho}\tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}} \succ \mathbf{0}$. Hence there must exist some $\delta > 0$ such that the three conditions in (A.30) hold true. \blacksquare

APPENDIX B

PROOF OF THEOREM 3

Proof of Theorem 3(a): Without loss generality, we assume that (P2) and problem (4) are unconstrained, i.e., $\mathcal{X}_i = \mathbb{R}^K \forall i \in V$, since they can be implicitly included in the nonsmooth component g_i 's [36, Section 5]. Let $(\mathbf{x}_1^*, \dots, \mathbf{x}_N^*)$ be an optimal solution to (P2), and let $(\mathbf{y}_1^*, \dots, \mathbf{y}_N^*)$ and $\{\mathbf{u}_{ij}^*, \mathbf{v}_{ij}^*, j \in \mathcal{N}_i\}_{i=1}^N$ be a pair of optimal primal and dual solutions to problem (22). Then they respectively satisfy the following optimality conditions

$$\mathbf{A}_i^T \nabla f_i(\mathbf{x}_i^*) + \partial g_i(\mathbf{x}_i^*) + \mathbf{E}_i^T \mathbf{y}^* = \mathbf{0}, i \in V, \tag{A.31}$$

$$\sum_{i=1}^N \mathbf{E}_i \mathbf{x}_i^* = \mathbf{q}, \tag{A.32}$$

$$\partial\varphi_i(\mathbf{y}_i^*) + \frac{1}{N}\mathbf{q} + \sum_{j \in \mathcal{N}_i} \mathbf{u}_{ij}^* + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^* = \mathbf{0}, \quad i \in V, \quad (\text{A.33})$$

$$\mathbf{y}^* \triangleq \mathbf{y}_i^* = \mathbf{y}_j^* \quad \forall j \in \mathcal{N}_i, i \in V, \quad (\text{A.34})$$

$$\mathbf{u}_{ij}^* + \mathbf{v}_{ji}^* = \mathbf{0} \quad \forall j \in \mathcal{N}_i, i \in V. \quad (\text{A.35})$$

where $\partial\varphi_i(\mathbf{y}_i^*) = -\mathbf{E}_i \hat{\mathbf{x}}_i^*$ given that $\hat{\mathbf{x}}_i^*$ is a maximizer to (4) with $\mathbf{y} = \mathbf{y}_i^*$ [37].

Firstly, it follows from (32) and (A.33) that

$$\begin{aligned} \mathbf{0} &= -(\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N}\mathbf{q}) + \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} + \mathbf{v}_{ji}^{(k+1)}) \\ &\quad + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)} - \mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)}) \\ &= -\mathbf{E}_i \hat{\mathbf{x}}_i^* + \frac{1}{N}\mathbf{q} + \sum_{j \in \mathcal{N}_i} \mathbf{u}_{ij}^* + \sum_{j \in \mathcal{N}_i} \mathbf{v}_{ji}^*. \end{aligned} \quad (\text{A.36})$$

By multiplying $\mathbf{y}_i^{(k)} - \mathbf{y}^*$ to the both sides of (A.36), we obtain

$$\begin{aligned} &\sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k+1)} + \mathbf{v}_{ji}^{(k+1)} - \mathbf{u}_{ij}^* - \mathbf{v}_{ji}^*)^T (\mathbf{y}_i^{(k)} - \mathbf{y}^*) \\ &\quad + c \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k)} + \mathbf{y}_j^{(k)} - \mathbf{y}_i^{(k-1)} - \mathbf{y}_j^{(k-1)})^T (\mathbf{y}_i^{(k)} - \mathbf{y}^*) \\ &\quad - (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*)^T \mathbf{E}_i^T (\mathbf{y}_i^{(k)} - \mathbf{y}^*) = \mathbf{0}. \end{aligned} \quad (\text{A.37})$$

Secondly, by the optimality of (34) and by (24), we have the following chain

$$\begin{aligned} \mathbf{0} &= \mathbf{A}_i^T \mathbf{s}_i^{(k-1)} + \partial g(\mathbf{x}_i^{(k)}) + \frac{1}{2|\mathcal{N}_i|} \mathbf{E}_i^T \left[\frac{1}{c} (\mathbf{E}_i \mathbf{x}_i^{(k)} - \frac{1}{N}\mathbf{q}) \right. \\ &\quad \left. - \frac{1}{c} \sum_{j \in \mathcal{N}_i} (\mathbf{u}_{ij}^{(k)} + \mathbf{v}_{ji}^{(k)}) + \sum_{j \in \mathcal{N}_i} (\mathbf{y}_i^{(k-1)} + \mathbf{y}_j^{(k-1)}) \right] \\ &\quad + \mathbf{P}_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) \\ &= \mathbf{A}_i^T \mathbf{s}_i^{(k-1)} + \partial g(\mathbf{x}_i^{(k)}) + \mathbf{E}_i^T \mathbf{y}_i^{(k)} + \mathbf{P}_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) \end{aligned} \quad (\text{A.38})$$

$$\begin{aligned} &= \mathbf{A}_i^T (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^{(k)}) + \mathbf{A}_i^T \mathbf{s}_i^{(k)} + \partial g(\mathbf{x}_i^{(k)}) \\ &\quad + \mathbf{E}_i^T \mathbf{y}_i^{(k)} + \mathbf{P}_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) \end{aligned} \quad (\text{A.39})$$

$$= \mathbf{A}_i^T \mathbf{s}_i^* + \partial g(\hat{\mathbf{x}}_i^*) + \mathbf{E}_i^T \mathbf{y}^*, \quad (\text{A.40})$$

where $\mathbf{s}_i^{(k)} = \nabla f_i(\mathbf{A}_i \mathbf{x}_i^{(k)})$, $\mathbf{s}_i^* = \nabla f_i(\mathbf{A}_i \hat{\mathbf{x}}_i^*)$ and the last equality is owing to the fact that $\hat{\mathbf{x}}_i^*$ is a maximizer to (4) with $\mathbf{y} = \mathbf{y}^*$. Multiplying both (A.39) and (A.40) with $\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*$ and combining the resultant equation with (A.37) further yields (A.41) on the top of the next page. Multiplying both (A.39)

and (A.40) with $\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*$ and combining the resultant equation with (A.37) further yields

$$\begin{aligned} & (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^{(k)})^T \mathbf{A}_i (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*) + (\mathbf{s}_i^{(k)} - \mathbf{s}_i^*)^T \mathbf{A}_i (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*) + (\partial g(\mathbf{x}_i^{(k)}) - \partial g(\hat{\mathbf{x}}_i^*))^T (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*) \\ & + (\mathbf{y}_i^{(k)} - \mathbf{y}_i^*)^T \mathbf{E}_i (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*) + (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)})^T \mathbf{P}_i (\mathbf{x}_i^{(k)} - \hat{\mathbf{x}}_i^*) = \mathbf{0}. \end{aligned} \quad (\text{A.41})$$

By summing (A.41) for $i = 1, \dots, N$, followed by applying results similar to (A.6) to (A.16), one obtains

$$\begin{aligned} & \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{\mathbf{M} + \frac{1}{2}\mathbf{P}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 + \frac{c}{2} \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_Q^2 \\ & \leq \|\mathbf{x}^{(k-1)} - \hat{\mathbf{x}}^*\|_{\frac{1}{2}\mathbf{P}}^2 + \frac{1}{c} \|\mathbf{u}^{(k)} - \mathbf{u}^*\|_2^2 + \frac{c}{2} \|\mathbf{y}^{(k-1)} - \tilde{\mathbf{y}}^*\|_Q^2 \\ & \quad - \frac{1}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_{\mathbf{P} - \frac{1}{\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2 - \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 \\ & \quad - \frac{c}{2} \|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\|_Q^2, \end{aligned} \quad (\text{A.42})$$

where $\mathbf{Q} \triangleq (\mathbf{D} + \mathbf{W}) \otimes \mathbf{I}_M$, $\rho > 0$, $\hat{\mathbf{x}}^* = [(\hat{\mathbf{x}}_1^*)^T, \dots, (\hat{\mathbf{x}}_N^*)^T]^T$, $\tilde{\mathbf{y}}^* = \mathbf{1}_N \otimes \mathbf{y}^*$, $\mathbf{y}^{(k)} = [(\mathbf{y}_1^{(k)})^T, \dots, (\mathbf{y}_N^{(k)})^T]^T$, $\mathbf{P} = \text{blkdiag}\{\mathbf{P}_1, \dots, \mathbf{P}_N\} \succ \mathbf{0}$, and \mathbf{D}_{σ_f} , \mathbf{D}_{L_f} , $\tilde{\mathbf{A}}$ and $\mathbf{M} = \tilde{\mathbf{A}}^T (\mathbf{D}_{\sigma_f} - \frac{\rho}{2} \mathbf{I}_{NK}) \tilde{\mathbf{A}}$ are defined below (A.10). It therefore follows from (A.42) that if

$$\sigma_{f,i}^2 - \frac{\rho}{2} > 0, \quad \mathbf{P}_i - \frac{L_{f,i}^2}{\rho} \mathbf{A}_i^T \mathbf{A}_i \succ \mathbf{0}, \quad \forall i \in V, \quad (\text{A.43})$$

then

$$\begin{aligned} \mathbf{x}^{(k)} & \rightarrow \hat{\mathbf{x}}^*, \quad \mathbf{u}^{(k)} \rightarrow \mathbf{u}^*, \\ \mathbf{x}^{(k)} & \rightarrow \mathbf{x}^{(k-1)}, \quad \mathbf{u}^{(k+1)} \rightarrow \mathbf{u}^{(k)} \end{aligned}$$

as $k \rightarrow \infty$. The result of $\mathbf{u}_{ij}^{(k+1)} \rightarrow \mathbf{u}_{ij}^{(k)}$ and (23a) imply that $\mathbf{y}_i^{(k)} \rightarrow \mathbf{y}_j^{(k)} \forall j \in \mathcal{N}_i, i \in V$, which, together with Assumption 1 implies that $\mathbf{y}^{(k)} \rightarrow \mathbf{1}_N \otimes \mathbf{y}_i^{(k)}$ for any $i \in V$. Since the Laplacian matrix $\mathbf{L}\mathbf{1}_N = \mathbf{0}$ [30], one obtains

$$\begin{aligned} & \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_Q^2 \rightarrow (\mathbf{1}_N \otimes (\mathbf{y}_i^{(k)} - \mathbf{y}^*))^T \mathbf{Q} (\mathbf{1}_N \otimes (\mathbf{y}_i^{(k)} - \mathbf{y}^*)) \\ & = (\mathbf{1}_N^T (\mathbf{D} + \mathbf{W}) \mathbf{1}_N) \|\mathbf{y}_i^{(k)} - \mathbf{y}^*\|_2^2 \\ & = (\mathbf{1}_N^T (2\mathbf{D} - \mathbf{L}) \mathbf{1}_N) \|\mathbf{y}_i^{(k)} - \mathbf{y}^*\|_2^2 \\ & = (2 \sum_{j=1}^N |\mathcal{N}_j|) \|\mathbf{y}_i^{(k)} - \mathbf{y}^*\|_2^2, \end{aligned} \quad (\text{A.44})$$

which, when combined with (A.42), further implies that $\mathbf{y}_i^{(k)} \rightarrow \mathbf{y}^* \forall i \in V$. Finally, recall (A.38) and (33) which is also valid for IDC-ADMM. We conclude that, as $k \rightarrow \infty$, $(\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_N^{(k)})$ and $(\mathbf{y}_1^{(k)}, \dots, \mathbf{y}_N^{(k)})$ will satisfy (A.31) and (A.32) asymptotically and converge to a pair of primal-dual optimal solution of (P2). ■

Proof of Theorem 3(b): Now we assume that $\phi_i(\mathbf{x}_i) = f_i(\mathbf{A}_i \mathbf{x}_i)$, $\mathcal{X}_i = \mathbb{R}^K$ for all $i \in V$, \mathbf{A}_i 's have full column rank and that \mathbf{E}_i 's have full row rank. Denote $\mathbf{r}^{(k)} \triangleq \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{\alpha \mathbf{M} + \frac{1}{2} \mathbf{P}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 + \frac{c}{2} \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_Q^2$ for some $\alpha > 0$. One can write (A.42) as follows

$$\begin{aligned} & \mathbf{r}^{(k)} + \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{(1-\alpha)\mathbf{M}}^2 + \|\mathbf{x}^{(k-1)} - \hat{\mathbf{x}}^*\|_{\alpha \mathbf{M}}^2 \\ & \leq \mathbf{r}^{(k-1)} - \frac{1}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_{\mathbf{P} - \frac{1}{\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2 \\ & \quad - \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 - \frac{c}{2} \|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\|_Q^2. \end{aligned}$$

Therefore, it suffices to show that, for some $\delta > 0$,

$$\begin{aligned} & \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{(1-\alpha)\mathbf{M}}^2 + \|\mathbf{x}^{(k-1)} - \hat{\mathbf{x}}^*\|_{\alpha \mathbf{M}}^2 \\ & + \frac{1}{2} \|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_{\mathbf{P} - \frac{1}{\rho} \tilde{\mathbf{A}}^T \mathbf{D}_{L_f} \tilde{\mathbf{A}}}^2 + \frac{1}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_2^2 \\ & + \frac{c}{2} \|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\|_Q^2 \geq \delta \mathbf{r}^{(k)}. \end{aligned} \tag{A.45}$$

Firstly, from (A.38) and (A.40), we have that (in the absence of g_i 's)

$$\begin{aligned} & \mathbf{A}_i^T (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^*) + \mathbf{E}_i^T (\mathbf{y}_i^{(k)} - \mathbf{y}_i^*) \\ & + \mathbf{P}_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)}) = \mathbf{0}. \end{aligned} \tag{A.46}$$

By applying (A.28) to (A.46) and by (9), one can show that, for some $\mu_1 > 1$,

$$\begin{aligned} & \|\mathbf{P}_i (\mathbf{x}_i^{(k)} - \mathbf{x}_i^{(k-1)})\|^2 \\ & \geq (1 - \mu_1) \|\mathbf{A}_i^T (\mathbf{s}_i^{(k-1)} - \mathbf{s}_i^*)\|^2 + (1 - \frac{1}{\mu_1}) \|\mathbf{E}_i^T (\mathbf{y}_i^{(k)} - \mathbf{y}_i^*)\|_2^2 \\ & \geq (1 - \mu_1) L_{f,i} \lambda_{\max}^2 (\mathbf{A}_i^T \mathbf{A}_i) \|\mathbf{x}_i^{(k-1)} - \hat{\mathbf{x}}_i^*\|_2^2 \\ & \quad + (1 - \frac{1}{\mu_1}) \lambda_{\min} (\mathbf{E}_i \mathbf{E}_i^T) \|\mathbf{y}_i^{(k)} - \mathbf{y}_i^*\|_2^2. \end{aligned} \tag{A.47}$$

Note that $\mathbf{D} + \mathbf{W} = 2\mathbf{D} - \mathbf{L} \preceq 2\mathbf{D}$ due to $\mathbf{L} \succeq \mathbf{0}$ [30]. Hence, it follows from (A.47) that

$$\begin{aligned} & \frac{c\delta}{2} \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_Q^2 \leq c\delta \|\mathbf{y}^{(k)} - \tilde{\mathbf{y}}^*\|_{\mathbf{D} \otimes \mathbf{I}_M}^2 \\ & \leq c\delta \tau_1 \|(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})\|_{\mathbf{P}^T \mathbf{P}}^2 + c\delta \tau_2 \|\mathbf{x}^{(k-1)} - \hat{\mathbf{x}}^*\|_2^2, \end{aligned} \tag{A.48}$$

where $\tau_1 = \max_{i \in V} \left\{ \frac{|\mathcal{N}_i|}{(1 - \frac{1}{\mu_1}) \lambda_{\min} (\mathbf{E}_i \mathbf{E}_i^T)} \right\} > 0$ and $\tau_2 = \max_{i \in V} \left\{ \frac{(\mu_1 - 1) \lambda_{\max}^2 (\mathbf{A}_i^T \mathbf{A}_i) |\mathcal{N}_i| L_{f,i}}{(1 - \frac{1}{\mu_1}) \lambda_{\min} (\mathbf{E}_i \mathbf{E}_i^T)} \right\} > 0$ are finite given that \mathbf{E}_i 's have full row rank.

Secondly, upon stacking (A.36) for all $i \in V$ and applying (A.3) and (A.11), one obtains

$$\begin{aligned} & \Psi(\mathbf{u}^{(k+1)} - \mathbf{u}^*) + c\mathbf{Q}(\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}) \\ & - \mathbf{E}(\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*) = \mathbf{0}, \end{aligned} \tag{A.49}$$

where $\mathbf{E} = \text{blkdiag}\{\mathbf{E}_1, \dots, \mathbf{E}_N\}$ and Ψ is given in (A.25). Analogously, by applying (A.28) to (A.49) and by (A.26), one can show that, for some $\mu_2 > 1$,

$$\begin{aligned} \frac{\delta}{c} \|\mathbf{u}^{(k+1)} - \mathbf{u}^*\|_2^2 &\leq \frac{\delta}{c\tau_3} \|\mathbf{x}^{(k)} - \hat{\mathbf{x}}^*\|_{\mathbf{E}^T \mathbf{E}}^2 \\ &\quad + \frac{\delta(\mu_2 - 1)c}{\tau_3} \|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\|_{\mathbf{Q}}^2, \end{aligned} \quad (\text{A.50})$$

where $\tau_3 = (1 - \frac{1}{\mu_2})\sigma_{\min}^2(\Psi) > 0$. By (A.48) and (A.50), sufficient conditions for satisfying (A.45) are therefore given by, $\forall i \in V$,

$$(1 - \alpha - \delta\alpha)(\sigma_{f,i}^2 - \frac{\rho}{2})\mathbf{A}_i^T \mathbf{A}_i \succeq \frac{\delta}{2}\mathbf{P}_i + \frac{\delta}{c\tau_3}\mathbf{A}_i^T \mathbf{A}_i, \quad (\text{A.51a})$$

$$\alpha(\sigma_{f,i}^2 - \frac{\rho}{2})\mathbf{A}_i^T \mathbf{A}_i \succeq c\delta\tau_2 \mathbf{I}_K, \quad (\text{A.51b})$$

$$\frac{1}{2}\mathbf{P}_i - \frac{L_{f,i}^2}{2\rho}\mathbf{A}_i^T \mathbf{A}_i \succeq c\delta\tau_1 \mathbf{P}_i^T \mathbf{P}_i, \quad (\text{A.51c})$$

$$\frac{1}{2} \geq \frac{\delta(\mu_2 - 1)}{\tau_3}. \quad (\text{A.51d})$$

Under (A.43), we see that (A.51) can be true for some $\delta > 0$. The proof is complete. \blacksquare

REFERENCES

- [1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in *Proc. Grid Computing Environments Workshop*, Austin, TX, USA, Nov. 12-16, 2008, pp. 1–10.
- [2] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up Machine Learning- Parallel and Distributed Approaches*. Cambridge University Press, 2012.
- [3] G. R. Andrews, *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2007.
- [4] S. Ghosh, *Distributed Systems- An Algorithmic Approach*. Chapman & Hall/CRC Computer & Information Science Series, 2007.
- [5] M. Elad, *Sparse and Redundant Representations*. New York, NY, USA: Springer Science + Business Media, 2010.
- [6] A. Nedić, A. Ozdaglar, , and A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” *IEEE Trans. Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010.
- [7] B. Johansson, T. Keviczky, M. Johansson, and K. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems,” in *Proc. IEEE CDC*, Cancun, Mexico, Dec. 9-11, 2008, pp. 4185–4190.
- [8] M. Zhu and S. Martínez, “On distributed convex optimization under inequality and equality constraints,” *IEEE Trans. Automatic Control*, vol. 57, no. 1, pp. 151–164, Jan. 2012.
- [9] J. Chen and A. H. Sayed, “Diffusion adaption strategies for distributed optimization and learning networks,” *IEEE. Trans. Signal Process.*, vol. 60, no. 8, pp. 4289–4305, Aug. 2012.
- [10] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
- [11] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, “Distributed basis pursuit,” *IEEE. Trans. Signal Process.*, vol. 60, no. 4, pp. 1942–1956, April 2012.
- [12] D. P. Bertsekas, *Network Optimization : Contribuous and Discrete Models*. Athena Scientific, 1998.

- [13] C. Shen, T.-H. Chang, K.-Y. Wang, Z. Qiu, and C.-Y. Chi, "Distributed robust multicell coordinated beamforming with imperfect CSI: An ADMM approach," *IEEE Trans. Signal Processing*, vol. 60, no. 6, pp. 2988–3003, 2012.
- [14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.
- [15] S. S. Ram, A. Nedić, and V. V. Veeravalli, "A new class of distributed optimization algorithm: Application of regression of distributed data," *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.
- [16] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," submitted to *IEEE Trans. Automatic Control*. Available on arxiv.org.
- [17] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.
- [18] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Dec. 2010.
- [19] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Stat. Soc. B*, vol. 58, pp. 267–288, 1996.
- [20] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Puschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 2718–2723, May 2013.
- [21] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Rice CAAM technical report 12-14, 2012.
- [22] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," available on arxiv.org.
- [23] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," available on arxiv.org.
- [24] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, 2005.
- [25] B. He and X. Yuan, "Linearized alternating direction method of multipliers with gaussian back substitution for separable convex programming," *Numerical Algebra. Control and Optimization*, vol. 3, no. 2, pp. 247–260, 2013.
- [26] S. Ma, "Alternating proximal gradient method for convex minimization," available on <http://www.optimization-online.org/>.
- [27] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Large-scale sparse logistic regression," in *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, June 28 - July 1, 2009, pp. 547–556.
- [28] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," available on arxiv.org.
- [29] B. He, Z. Peng, and X. Wang, "Proximal alternating direction-based contraction methods for separable linearly constrained convex optimization," *Frontiers of Math. in China*, vol. 6, no. 1, pp. 79–114, 2011.
- [30] F. R.-K. Chung, *Spectral graph theory*. CBMS Regional Conference Series in Mathematics, No. 92, 1996.
- [31] M. E. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3753–3764, 2008.
- [32] J. Baliga, R. Ayre, and R. S. Tucker, "Energy consumption in wired and wireless access networks," *IEEE Commun. Mag.*, pp. 70–77, June 2011.
- [33] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1996.
- [34] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [35] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 1–112, 2013.
- [36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[37] S. Boyd and A. Mutapcic, “Subgradient methods,” available at www.stanford.edu/class/ee392o/subgrad_method.pdf.